młody TECHNIK

InforMik

MAGAZYN KOMPUTEROWY „MŁODEGO TECHNIKA"



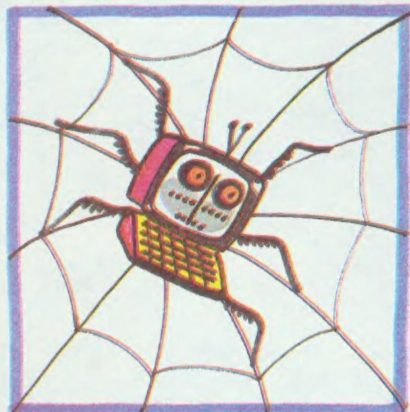
Nasz komentarz:

PAJĘCZYNA

Jak wiadomo rynek komputerowy w Polsce stał się dziś rynkiem klienta. Mimo iż popyt ciągle rośnie (tendencji tej nie zmieniają chwilowe przyhamowania, spowodowane meandrami reformy gospodarczej), podaż przewyższa go istotnie. Komputery i drukarki dziś oferują prawie wszyscy, gdyż handel nimi jest mimo wszystko sprawą stosunkowo prostą. Konkurencja między sprzedawcami staje się coraz ostrzejsza, do historii należą kilkudziesięcioprocentowe marże pośredników. Normą stała się już marża w wysokości 5%, oczywiście w przypadku, gdy klient rezygnuje z gwarancji. Jednym słowem — rynek dojrzewa. Nie znaczy to jednak, że jest to rynek całkiem zdrowy.

Firmy handlowe, chcąc utrzymać się na powierzchni, szukają różnych sposobów pozyskania klienta. Niektóre sięgają po sposoby mało ambitne, ale sprawdzone. Gdy w firmie pojawi się przedstawiciel przedsiębiorstwa chętnego do nabycia jakiegoś sprzętu, zanim jeszcze przedstawi on swe życzenia, już otrzyma propozycję podpisania umowy o akwizycję. W razie zawarcia transakcji firma odpali „akwizytorowi" stosowną dolę. Jak widać, w naszym kraju dopracowaliśmy się już nawet legalnych form wręczania łapówek.

Inne firmy próbują zwiększać obroty przez odkrywanie luk rynkowych. Zamówienia na specjalizowane urządzenia peryferyjne są zwykle lukratywne, ale też niosą za sobą większe ryzyko i kłopoty, zwłaszcza w razie konieczności zapewnienia serwisu. Od pewnego czasu dużym powodzeniem u klientów cieszą się też sieci komputerowe, co odpowiada zresztą światowym tendencjom. W Polsce zapotrzebowanie na sieci ma dodatkowe podłoże: komputery osobiste mają zneutralizować brak dużych, wielostanowiskowych systemów komputerowych. W związku z powyższym liczni pośrednicy włączają do swojej oferty różnego rodzaju karty sieciowe i inne akcesoria do budowy sieci lokalnych w oparciu o komputery klasy PC/XT i AT.



Handel sieciami pozwala, jak sądzę, dostrzec zjawiska, które niedługo staną się w branży komputerowej znacznie powszechniejsze. Otóż właśnie przy handlu sieciami wielu zwabionych zyskami pośredników przekroczyło swój próg niekompetencji. Instalacja sieci jest z natury przedsięwzięciem dość trudnym i wymagającym dobrego rozpoznania. Jedne sieci obejmują jedno pomieszczenie lub kilka sąsiednich pokoi, inne znów rozciągają się na terenie całego zakładu, a odległości między poszczególnymi stacjami są kilometrowe. Jedne sieci łączą ze sobą 2—3 komputery, podczas gdy w innych planuje się docelowo współpracę kilkuset stacji. Różne są też wymagania dotyczące dostępu do poszczególnych zasobów w sieci i tras przepływu danych.

Nie tylko topologia wpływa na wybór rozwiązania sieciowego. Bardzo istotne jest przewidywane oprogramowanie. W znacznej większości sieci prędzej czy później wystąpi problem wielodostępu, tzn. równoczesnego korzystania z tych samych plików danych przez wielu użytkowników, z których część może mieć ochotę modyfikować zawartość tych plików. Nie każde oprogramowanie jest dostosowane do pracy w stwarzanym przez sieć środowisku wielodostępnym. Co gorsza, istnieje kilka różnych mechanizmów realizacji wielodostępu — różne sieci udostępniają różne mechanizmy. Mnogość czynników sprawia że właściwy dobór sieci jest bardzo złożony, a jego ogarnięcie przekracza możliwości Pana Ziutka, zatrudnianego przez wiele firm jako fachowca „od komputerów". W wolnych chwilach Pan Ziutek obsługuje w sklepie wideo, nastraja kolorowy telewizor albo nastawia antenę satelitarną.

W powyższej sytuacji przypadkowo dobrane karty sieciowe leżą na sklepowych półkach jak kartofle, a cały ciężar wyboru spoczywa na kliencie. Nic dziwnego, że w tej sytuacji procent nie trafionych inwestycji sieciowych okazuje się dość znaczny. Każda sieć po zainstalowaniu pozwala przesyłać pliki między komputerami, dostarczając nabywcy pierwszych radosnych przeżyć. Problemy pojawiają się dopiero później. Na dyskach znikają pliki albo ulegają cyklicznym uszkodzeniom cenne dane. Transmisja danych wykazuje okresowe, trudne do wyjaśnienia błędy. Od czasu do czasu z niewiadomych powodów „padają" karty sieciowe.

Sprawna i sensowna organizacja sieci wymaga fachowców o odpowiednim przygotowaniu, doświadczeniu i wyobraźni. Ludzi, którzy potrafią sensownie dobrać topologię sieci na podstawie nie zawsze sprecyzowanych planów, a często pobożnych życzeń klienta. Specjalistów ogarniających związki między siecią a pracującym w niej oprogramowaniem. Ekspertów rozumiejących zagrożenia związane z galwanicznym łączeniem komputerów, złym uziemieniem poszczególnych stacji, prądami błądzącymi, wyładowaniami atmosferycznymi w sąsiedztwie przewodów sieci biegnących pomiędzy budynkami, szczurami grasującymi w kanałach kablowych. Zwykłej firmy handlowej nie stać na stałe zatrudnienie takiego fachowca, a klient albo nie ma do niego dostępu, albo wcale nie odczuwa potrzeby skorzystania z jego usług w fazie zakupu. Po pomoc fachowców sięga się dopiero wtedy, gdy pojawiają się nierozwiązywalne problemy i sprzeczności. Niestety, często konkluzja rzeczoznawcy sprowadza się do stwierdzenia, że pieniądze wyrzucono w błoto, gdyż sieć nie spełnia elementarnych warunków. Kiedy indziej okazuje się, że zakupiono drogą sieć, a potrzeby są tego rzędu, że spełniłaby je zwyczajna sieć oparta na standardowych interfejsach RS232C, jak np. popularny LANLINK.

A propos LANLINKA: ta niedoceniana sieć lokalna ma sporo zalet i doskonale nadaje się do pierwszych eksperymentów z sieciami, które pozwolą nabrać niezbędnego doświadczenia i rozeznania w problemach środowiska sieciowego. Aż dziw bierze, jak wielu klientów porywa się na kosztowne inwestycje nie próbując nawet przećwiczyć swych problemów w modelowych warunkach w oparciu np. o nie wymagającą praktycznie żadnych inwestycji sieć LANLINK!

Stopień technicznego skomplikowania w zestawieniu z małym z reguły kalibrem firm handlowych narzuca konieczność korzystania z usług wyspecjalizowanych firm konsultingowych. Poza doradztwem technicznym firmy takie mogłyby zresztą świadczyć inne usługi w rodzaju szkolenia personelu klienta, analizy systemów informatycznych, wdrażania oprogramowania itd. Informatycznych przedsiębiorstw konsultingowych jest ciągle jak na lekarstwo, a niektóre istniejące są nimi tylko z nazwy, koncentrując swą działalność na handlu. Handel rzeczami wciąż opłaca się bardziej, niż handel myślą techniczną i organizacyjną. Właściwe urzędy także nie zasypiają gruszek w popiele. Jak skalkulować tę myśl techniczną? Jak ją opodatkować? Jak w trakcie kontroli ustalić, ile tej myśli zalega w magazynie, ile w formie półwyrobów, ile w postaci gotowych produktów?

Klientów, zainteresowanych kompleksowymi usługami konsultingowymi, jest coraz więcej. Niektórzy przychodzą jednak po radę już z poparzonymi rękami. Często dotyczy to użytkowników sieci lokalnych, dla których sieć taka niepostrzeżenie przerodziła się w koszmarną pajęczynę.

Roland Wacławek

młody
TECHNIK
InforMik

NR IV [8] ROCZNIK II

SPIS TREŚCI

Zdjęcia w numerze: Władysław P. Jabłoński, Tadeusz Rzepecki.
Rysunki: Jerzy Flisak, Roman Gaik.

FELIETON PRZECIW FELIETONISTOM



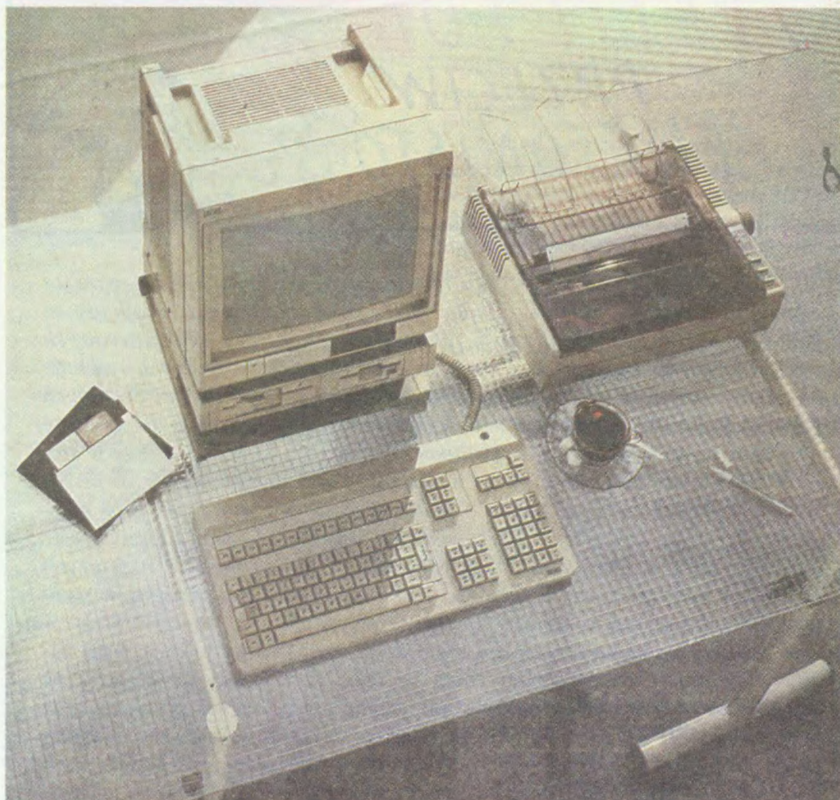
W epoce ucieczki od wysiłku umysłowego, scedowanego na myślące maszyny, także i rozrywka stała się sztuką bezmyślną. Najlepszym dowodem na to jest popularność filmów tzw. akcji, gier komputerowych i muzyki typu np. italo-disco. Co gorsze i tzw. rozrywka umysłowa wydatnie obniżyła swoje loty — być może dlatego, że obecnie „Kabaret Starszych Panów" byłby nie do przyjęcia, gdyż jego Autorzy kierowali go do odbiorcy wrażliwego, inteligentnego i wyrobionego towarzysko przynajmniej w sferach zbliżonych do high life'u (kto dziś wie bez zaglądania do słownika wyrazów obcych co to są: apanaże, ordewry itp.). Nie znaczy to, broń mnie Panie Boże, że jesteśmy społeczeństwem idiotów — wręcz przeciwnie: jesteśmy bardzo wyczuleni, a nawet czasem przeczuleni w niektórych obszarach naszego życia, jednak rozrywkę lubimy prostą, przaśną, w stylu operujących tortami komedii z lat dwudziestych.

Jedną z najciekawszych dla mnie form rozrywki umysłowej była zawsze felietonistyka — piszący facet musi co jakiś czas zrobić swojemu czytelnikowi wodę z mózgu, aby **a)** *udowodnić wyższość własnych szarych komórek nad tymiż u czytelnika,* **b)** *pokazać, ża dana sprawa nie wygląda wcale tak prosto, jak się nam wydaje i właśnie dzięki* **a)** *będzie nam odsłonięte jej drugie dno, oraz* **c)** *zareagować szybko na aktualne i uniwersalne problemy czytelnika tłumacząc mu jak glupawemu dziecku, że np. po ciemku nawet białe jest czarne. Ktoś kiedyś zrobił błyskotliwą uwagę, że dobry felietonista powinien przypominać foksteriera: być ciętym i agresywnym oraz dużo szczekać...*

Jaki związek ma powyższy wywód z komputerami? Ano taki, że niegdyś niemal każdy felietonista sięgał po przykład z komputera, a w najgorszym razie rzucał jakimś magicznym słówkiem „z branży", nawoływał do komputeryzacji kraju i gospodarki lub pokazywał komputerowe przykłady spoza granic dla poparcia swoich wywodów. Moda minęła, dziennikarskich portable'ów wśród felietonistów za dużo chyba nie ma, a w każdym razie nie są one źródłem natchnienia swoich posiadaczy i temat znikł z łamów prasy, a jeśli wraca, to tylko w formie dziennikarskiej czkawki w sezonie ogórkowym.

A przecież skromna prasa fachowa czy hobbistyczna z tej branży nie przełamie oporów w gospodarce i społeczeństwie, również i dlatego, iż oportuniści posądzają ją o interesowność. Aby zwalczyć nasz deficyt rąk do pracy (ciekawe — w sklepach i komunikacji miejskiej tłok, a brak nam podobno ponad 800 000 pracowników) komputeryzacja jest PRZYMUSEM, jeśli nie chcemy rozsypać się z hukiem jako społeczeństwo, które w dobie obecnej łączy przede wszystkim wspólnota organizacyjna życia codziennego (oczywiście zaraz po wspólnocie terytorialnej!). W tym właśnie mają pomóc komputerom felietoniści, którzy jako ludzie inteligentni powinni widzieć ostrzej opisaną wyżej konieczność. Ale nie — oni wolą pisać o niepokojach społecznych, powinnościach rządu czy nowej inicjatywie międzynarodowej. To na pewno łatwiejsze i bezpieczniejsze, bo z perspektywy nic człowiekowi nie zleci na głowę, a komputer — rzecz skomplikowana i, jak pisał hr. Fredro, „tam zalotów, tam namowy, tam potrzeba tęgiej głowy". No cóż, jak pisałem wyżej, Polacy to naród zdolny i już niestety odkryli, że najlepszy pieniądz, to ten co nie śmierdzi ludzkim potem...

JERZY KLAWIŃSKI



ROLAND WACŁAWEK

DESKTOP PUBLISHING:

Jak się to robi?

Hasło: *Desktop Publishing* (w Polsce lansowana jest obecnie nazwa: *pulpit wydawniczy*) robi ostatnio zawrotną karierę.

Oznacza to przełamanie kolejnej bariery technologicznej i udostępnienie użytkownikowi zwykłego komputera osobistego takich narzędzi, jakie dotychczas były zarezerwowane jedynie dla profesjonalnej poligrafii o bardzo kosztownym uzbrojeniu technicznym.

Angielski termin: *Desktop Publishing* odnosi się do systemów mikrokomputerowych, z zasady jednostanowiskowych, pozwalających na kompleksowe redagowanie publikacji na poziomie zbliżonym do profesjonalnych wydawnictw (w polskich warunkach: na poziomie przewyższającym wiele profesjonalnych wydawnictw). Wszelkie operacje redakcyjne odbywają się w trybie dialogu z komputerem, nie występuje ostry rozdział czasowy poszczególnych faz pracy, jak pisanie tekstów, jego składanie, korekta, łamanie, montaż, itd. Fazy te przeplatają się, tak że można np. dokonać zmian w tekście na zmontowanych już stronicach nawet na 5 minut przed drukiem. Wydruk publikacji może odbywać się na własnej drukarce systemu — np. mozaikowej lub laserowej — albo też dostarczany jest opis publikacji w formie bezpośrednio akceptowalnej przez stosowane w drukarniach elektroniczne systemy fotoskładu.

Zastosowanie komputerów osobistych do wydruku wydawnictw nie jest niczym nowym. Przetwarzanie tekstów było jednym z ich głównych zastosowań od czasów Apple I. Jakież więc nowe właściwości wnosi pulpit wydawniczy w porównaniu z typowymi, w międzyczasie dość już wyrafinowanymi, edytorami tekstu? Próba porównania pulpitu wydawniczego z edytorem, choć pozornie logiczna, a nawet narzucająca się, jest jednak nieporozumieniem. Dlaczego?

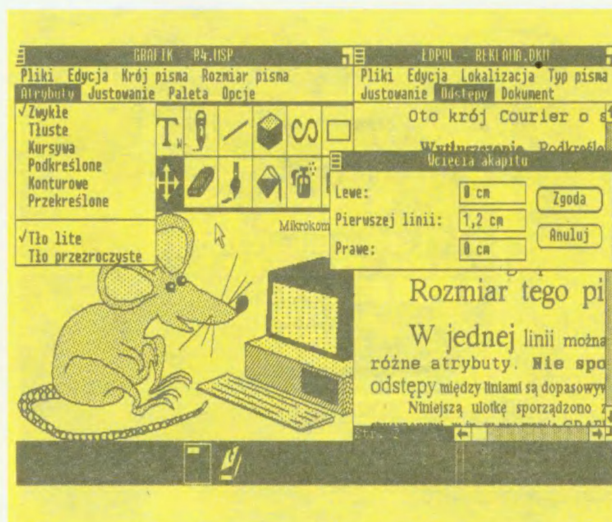
Głównym zadaniem edytora jest sprawne wprowadzanie, korekta i formatowanie tekstów o różnej objętości i strukturze. Tekst ma być drukowany możliwie szybko i na dowolnej drukarce, tak rozetkowej jak mozaikowej, co już z góry ogranicza swobodę operowania liternictwem ze względu na skromne możliwości takich drukarek. W edytorach tekstu główny nacisk kładzie się na elektroniczne manipulowanie tekstem, z mniej lub bardziej drugorzędnym traktowaniem jego formy graficznej po wydruku.

„Prawdziwa" publikacja to jednak o wiele więcej niż sam tylko tekst, chociażby najzręczniej opracowany. Publikacja jest bowiem dziełem o określonych walorach estetycznych. W najprostszym przypadku dotyczy to liternictwa i kompozycji tekstu z uwzględnieniem odpowiedniej liczby szpalt, nagłówków, tytułów itd. W praktyce większość publikacji zawiera także ilustracje oraz różnego rodzaju ornamenty i liniaturki (linie, ramki i inne kontury okalające lub oddzielające partie tekstu), płaszczyzny o różnym stopniu szarości „podłożone" pod tekst i ilustracje, itd. Tak więc publikacja jest dziełem graficznym, którego estetyka musi być ukształtowana narzędziami o całkiem innych właściwościach niż zwykły edytor tekstu. Edytory, i owszem, służą do produkcji ważnego elementu merytorycznej zawartości publikacji, jakim są teksty. Redagowanie tekstu odbywa się jednak we wcześniejszej fazie prac niż ostateczne projektowanie publikacji. Nawiązując do konwencjonalnej techniki wydawniczej, edytor jest narzędziem dla autora, natomiast pulpit wydawniczy — dla redaktora technicznego, a poniekąd także dla drukarza.

Co umożliwia typowy pulpit wydawniczy? Oto jego podstawowe funkcje:

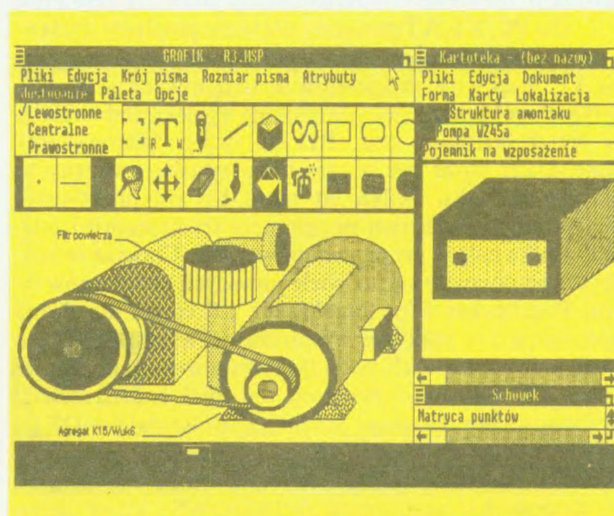
— Komponowanie tekstu w jednej lub kilku szpaltach z możliwością zmiennej liczby szpalt na stronie i precyzyjnym wyrównaniem marginesów.
— Dobór kroju, rozmiaru i innych atrybutów czcionki spośród przynajmniej kilku typów pisma proporcjonalnego.
— Swobodne definiowanie odstępów między szpaltami, liniami a nawet wyrazami i literami tekstu.
— Integrowanie w jednej publikacji tekstów napisanych za pomocą różnych edytorów tekstu, korekta i formatowanie tych tekstów, a także wprowadzanie nowych tekstów z klawiatury.
— Włączanie do publikacji ilustracji sporządzonych za pomocą różnych programów graficznych (np. Paintbrush czy PC-Paint w przypadku rodziny IBM PC/XT i AT), programów wspomagających projektowanie (np. AutoCAD), grafiki prezentacyjnej (np. Framework, Lotus) oraz grafiki uzyskanej w wyniku analizy różnych dokumentów (zdjęć, map, szkiców) za pomocą skanera lub np. analizatora obrazu TV.
— Uzupełnianie i korekta ilustracji, w tym zmiana rozmiarów i proporcji oraz kadrowanie: obcinanie, względnie maskowanie zbędnych fragmentów.
— Rysowanie elementów liniaturkowych i wypełnionych płaszczyzn (o różnym kształcie i wypełnieniu).
— Prezentacja gotowej publikacji na ekranie, strona po stronie, w postaci maksymalnie zbliżonej do przyszłego wydruku (tzw. zasada What You See Is What You Get, czyli WYSIWYG).
— Wydruk publikacji na różnych drukarkach, ale z możliwością pełnego wykorzystania zaawansowanych możliwości typograficznych i rozdzielczości drukarek wysokiej klasy, zwłaszcza laserowych, a także zapis publikacji w specjalizowanych językach opisu stronicy, jak PostScript.

Można spotkać opinie, że sensowne wykorzystanie pulpitu wydawniczego jest możliwe wyłącznie pod warunkiem posiadania drukarki laserowej. Osobiście sądzę, że pochodzą one głównie od osób, które gorliwie czytają zachodnią prasę fachową, ale mniej gorliwie praktykują



przy komputerze. Jest faktem, że drukarka laserowa gwarantuje uzyskanie wydruku wysokiej jakości, o rozdzielczości i obrazie czcionki zbliżonym do osiągalnego w typowej drukarni. Taką samą korzyść uzyskamy jednak także, drukując na drukarce laserowej teksty ze zwyczajnego edytora. W moim przekonaniu głównym walorem pulpitu wydawniczego jest jednak uzyskanie zupełnie nowego środowiska roboczego do samego projektowania publikacji, łatwość integrowania danych, znaczne zwiększenie swobody kompozycji i komfortu pracy oraz wzbogacenie warsztatu o szereg nowych, nie spotykanych w zwykłych edytorach tekstu środków wyrazu. Walory te mogą być wykorzystane niezależnie od posiadanej drukarki. Na popularnej drukarce mozaikowej techniczna jakość wydruku będzie niższa niż na laserowej, ale pozostałe walory estetyczne, związane z kompozycją publikacji, zostaną przecież zachowane. Brak porównywalnego wyposażenia technicznego z bogatymi krajami zachodnimi nie powinien, jak sądzę, powodować obrażania się na nową technologię, ale skłaniać nas do wykorzystania jej w taki sposób, na jaki nas stać. Na marginesie: rozwój drukarek — i laserowych, i mozaikowych — ciągle trwa, ceny spadają, a parametry rosną.

Problem rozdzielczości można złagodzić za pomocą wybiegu: zaprojektować i wydrukować publikację na drukarce mozaikowej w większej skali, a następnie po-





mniejszyć ją przy powielaniu, czemu odpowiada względny wzrost rozdzielczości.

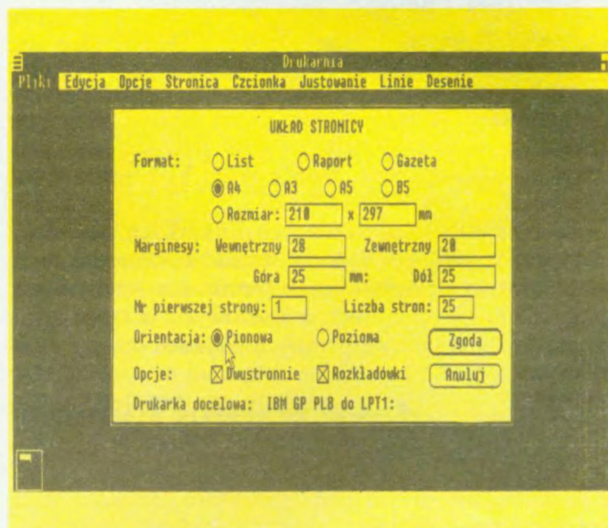
Oprogramowanie pulpitu wydawniczego jest dość złożone i obszerne, a realizowane operacje wymagają wydajnego procesora. Dużą objętość mają też przetwarzane zbiory danych. Dlatego też jest ono dostępne tylko dla komputerów osobistych o dużej pamięci operacyjnej, wyposażonych w pamięć masową o dostępie swobodnym, najlepiej — dysk twardy. Spośród spotykanego w kraju sprzętu w rachubę wchodzą więc w zasadzie dwie rodziny: IBM PC/XT/AT i Atari ST. Przyjrzyjmy się procesowi redagowania publikacji na przykładzie programu PageMaker 1.0 firmy Aldus dla PC/XT/AT.

Dla PC/XT/AT powstało już wiele pulpitów wydawniczych. Niektóre, bardziej zaawansowane, działają tylko na AT. Większość programów wymaga twardego dysku, w tym PageMaker, choć zdarzają się wyjątki, zadowalające się dyskietkami (np. ByLine firmy Ashton-Tate). PageMaker może być w tym towarzystwie uważany za klasyka gatunku. Kilka lat temu był on praktycznie pierwszym popularnym programem tej kategorii i odegrał wielką rolę w propagacji nowej idei. Chociaż w międzyczasie pojawiło się już wiele nowszych programów (w tym PageMaker 3.0), PageMaker 1.0 ciągle ma dobre notowania i w praktyce wystarcza do wszystkich typowych prac edytorskich, poza tym wystarcza mu komputer klasy XT. Dla mnie program ten miał istotną zaletę: dysponowałem do niego nakładką spolszczającą DRUKARNIA, zapewniającą m.in. polskie liternictwo na ekranie i na drukarce. Oprócz tego program ten pracuje w środowisku graficznym WINDOWS, które samo w sobie zawiera kilka przydatnych narzędzi pomocniczych, a zwłaszcza ułatwia import danych z innych programów — w tym metodą kopiowania ich wprost z ekranu. Bzdurny jest też kolportowany przez prasę pogląd, jakoby program ten nie nadawał się do redagowania publikacji o objętości przekraczającej 10 stronic. Zdarzyło mi się sprawnie i bez żadnych problemów zmajstrować za jego pomocą „cegłę" liczącą 115 stronic formatu A4.

Od czego zaczyna się praca nad publikacją? Oczywiście od opracowania jej ogólnej koncepcji. Przypuśćmy, że chciałbym stworzyć obficie ilustrowaną broszurę — instrukcję obsługi i konserwacji pewnego urządzenia technicznego, liczącą ok. 30 stronic formatu A4. Zanim uruchomię pulpit wydawniczy, przygotuję komplet ilustracji i przynajmniej podstawowe teksty. Co prawda mógłbym wpisywać teksty wprost w szpalty już na pulpicie, ale lepiej jest to zrobić za pomocą edytora, pozwalającego skoncentrować się na samej treści. Pozostaje mi więc założyć na dysku katalog roboczy, w którym będę gromadzić składniki przyszłej publikacji i zakasać rękawy.

Jeżeli chodzi o ilustracje, zachodzą dwa przypadki. Część planowanych obrazków to schematy i proste rysunki poglądowe. Narysuję je „od ręki" za pomocą jakiegoś odpowiedniego programu graficznego, jak np. In-A-Vision, Paintbrush albo Windows-Paint. Każdy z tych rysunków zostanie zapisany w osobnym pliku dyskowym. Gdyby zachodziła potrzeba włączenia do mojej ilustracji zdjęć, wówczas za pomocą skanera dokonałbym ich analizy, polecając zapisać je na dysku w formacie któregoś z rozpowszechnionych programów graficznych, jak np. któryś z powyższych. Dzięki temu mógłbym dokonać na zdjęciach niezbędnych korekt i retuszów, wprowadzić opisy itd. jeszcze przed włączeniem obrazka do mojej publikacji.

Pozostałe ilustracje będą selektywnymi wyciągami z dokumentacji technicznej urządzenia. Ponieważ wszelkie rysunki były tworzone za pomocą programu AutoCAD, sprawa jest dość prosta. Uruchamiam AutoCAD, kolejno ładuję pliki poszczególnych rysunków dokumentacji, wybieram z nich potrzebne fragmenty i uruchamiam funkcję kreślenia rysunku na ploterze. Jako urządzenie wyjściowe wybieram ADI, kierując równocześnie wydruk



nie do któregoś z fizycznych interfejsów, ale do pliku dyskowego. Otrzymany plik wydruku w formacie ADI jest niczym innym jak opisem poszczególnych wektorów, „kresek" rysunku. Plik ten może być później bezpośrednio wykorzystany przez PageMakera. Wybierając fragmenty rysunków do broszury nie muszę silić się na szczególną precyzję: lepiej wybrać fragment z naddatkiem, a ostatecznego kadrowania dokonać dopiero na pulpicie wydawniczym.

Mogłoby się wydawać, że taka pośrednia metoda wydobywania ilustracji z systemu AutoCAD jest niewygodna i niepotrzebnie skomplikowana. Czy PageMaker nie mógłby czytać rysunków (pliki z rozszerzeniem DWG) wprost, bez wydruku do pliku? Pozory mylą — w istocie metoda ta ma istotne zalety. Przed kreśleniem AutoCAD pozwala bowiem zrealizować wiele dodatkowych operacji: wybrać punkt obserwacji, ukryć obiekty niewidoczne z bieżącego punktu obserwacji, wyeliminować obiekty nieistotne, zawarte na wyodrębnionych planach, itd. Przygotowując rysunek można zatem wykorzystać cały bogaty arsenał środków graficznych systemu AutoCAD, co ze względu na potencjalną złożoność rysunków w tym systemie jest sprawą niebagatelną.

Kiedy gros materiału jest już gotowe, uruchamiam pulpit wydawniczy i wybieram w menu polecenie utworzenia nowej publikacji. Na samym wstępie muszę poczynić pewne założenia, dotyczące formatu publikacji: rozmiarów stronicy, szerokości marginesów, przewidywanej liczby stronic, a nawet typu drukarki i przewidywanego sposobu powielenia. W publikacji powielanej jednostronnie wszystkie stronice są „prawe". W publikacji dwustronnej rozróżniamy stronice lewe i prawe.

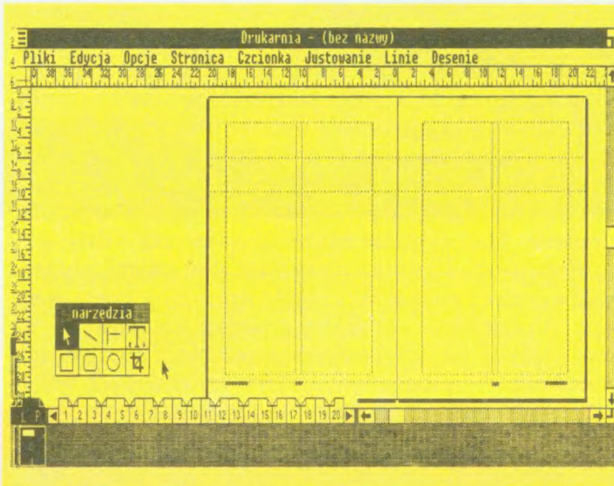
W systemie PageMaker każda stronica publikacji składa się z szeregu warstw-planów, na podobieństwo arkuszy przezroczystej folii, ułożonych jedna na drugiej. Na każdym z tych arkuszy może znajdować się rysunek albo blok tekstu. Poszczególne plany-arkusze można dowolnie przesuwać względem siebie, a także zmieniać kolejność ich ułożenia. Obiekty złożone na warstwach wierzchnich przesłaniają obiekty położone głębiej. Każda stronica ma swój własny komplet warstw. Istnieje jednak blok folii, który można nakładać na każdą stronicę. Są to tzw. szablony stronic. Zawierają one wszystkie te elementy, które powtarzają się na wszystkich stronicach. Może to być np. numer stronicy, nagłówki i stopki, stałe ornamenty, a także siatka traserska.

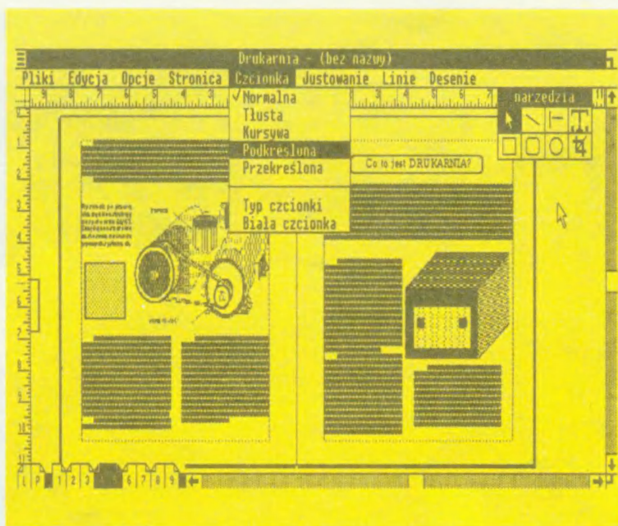
Siatka traserska jest zbiorem pionowych i poziomych linii pomocniczych, widocznych na ekranie i ułatwiających rozmieszczanie tekstu i ilustracji, ale niewidocznych na wydruku. Siatka ta składa się z marginesów, ograniczających roboczy obszar stronicy, prowadnic szpalt (pionowych linii wytyczających granice szpalt tekstu) oraz pionowych i poziomych linii odniesienia podziałki, pozwalających precyzyjnie odmierzać odległości i „zgrywać" ze sobą sąsiednie obiekty. Dostępny jest też tzw. tryb wiązania do siatki, w którym wystarczy umieścić jakiś obiekt, np. blok tekstu lub ilustrację, w bliskim sąsiedztwie linii pomocniczej, aby linia ta „złapała" krawędź obiektu. W ten sposób nie wysilając się specjalnie można doprowadzić do idealnego wyrównania krawędzi kilku obiektów.

Moją następną czynnością będzie właśnie projektowanie szablonów. Zaczynam od siatki traserskiej, decydując się na układ dwuszpaltowy, wstawiam oznacznik paginacji (przy wydruku program wpisze tam bieżący numer stronicy), wpisuję stopki stronicy. Teraz mogę już przystąpić do importowania z dysku poszczególnych składników treści. W zasadzie należałoby zacząć od rozmieszczania ilustracji, a następnie oplatać je tekstem. Ponieważ jednak zależy mi na ścisłym związku rysunków z komentarzem, będę plasował ilustracje na przemian z rozmieszczanym tekstem.



PageMaker dysponuje znacznie bogatszym repertuarem krojów pisma, niż typowe edytory. Zanim zacznę ładować teksty, wybiorę z menu standardowy krój pisma, który zostanie przypisany tym tekstom. Decyduję się na krój Times Roman (stosowany m.in. w wielu gazetach) o wysokości czcionki 12 punktów, zapewniającej dobrą czytelność (1 punkt typograficzny = 1/72 cala). Parametry czcionki będę mógł oczywiście zmieniać i później. Następnie „rozkładam" na pulpicie pierwszą stronicę, wybieram w katalogu dyskowym pierwszy plik tekstowy i wydaję rozkaz importu danych. Gdy PageMaker przeformatował już załadowany tekst zgodnie z wybranymi parametrami, co sygnalizuje zmiana kształtu kursora, wskazuję kursorem (za pomocą myszki) lewy górny róg pierwszej, na razie pustej szpalty i naciskam przycisk.





Tekst „wlewa" się w obszar ograniczony prowadnicami szpalt i zatrzymuje na linii dolnego marginesu.

Cały tekst, zaimportowany z pojedynczego pliku i stanowiący z reguły logicznie powiązaną całość, np. artykuł, tworzy tzw. wątek. Przy rozmieszczaniu na stronicach wątek jest dzielony pomiędzy bloki tekstowe. Blok tekstowy jest prostokątnym obszarem o dowolnie kształtowanych rozmiarach, posiadającym oprawkę i traktowanym jako całość przy przemieszczaniu itd. Oprawka bloku ma u góry i u dołu krawędzie z uchwytami.

Wyobraźmy sobie wątek jako ciecz, a bloki tekstowe jako zbiór elastycznych zbiorników, połączonych szeregowo przewodami. „Ciekły" wątek wypełnia układ naczyń połączonych. Ostatnie naczynie łączy się z rezerwuarem, zawierającym tę część wątku, dla której zabrakło miejsca w zbiornikach. Gdy powiększymy rozmiary któregoś z bloków-zbiorników, część wątku opuści rezerwuar i zajmie zwolnione miejsce. Ruch wątku obejmie wszystkie bloki między rezerwuarem, a miejscem zmian, tak że kolejność rozmieszczenia poszczególnych części wątku będzie zachowana, zmieni się natomiast sposób jego podziału między bloki.

Oprócz zmiany rozmiarów, w dowolnym miejscu szeregu można wstawić nowy blok albo usunąć blok istniejący, co spowoduje „przetłoczenie" zawartej w tym bloku części wątku do innych bloków. Podobnie, w dowolnym bloku można „dolać" nową część wątku (dopisanie tekstu) lub „odessać" część zbędną (skasowanie tekstu). Operacje te powodują przepływ wątku między poszczególnymi blokami i rezerwuarem. Mimo ruchu wątku, kontur zajmowanego przezeń obszaru w publikacji nie ulegnie zmianie, dopóki wprost nie zmienimy rozmiarów i rozmieszczenia bloków, wskazując kursorem ich odpowiednie uchwyty i wykonując odpowiednie manipulacje. Ma to istotne znaczenie zwłaszcza wtedy, gdy np. dopisujemy do wątku nowy tekst albo zmieniamy krój bądź rozmiar czcionki, co z reguły wpływa na zajętą przez tekst powierzchnię. Problem powstaje co najwyżej na końcu wątku, gdy trzeba dodać nowe bloki tekstowe lub przeciwnie — usunąć część bloków opróżnionych wskutek zmniejszenia obszaru zajętego przez tekst.

Ważną sprawą jest zapewnienie poprawnego dzielenia wyrazów. Jest on istotny zwłaszcza przy wąskiej szpalcie. Co dłuższe wyrazy, nie mieszczące się w wierszu, trzeba podzielić na dwie części, zaopatrując pierwszą z nich w łącznik na końcu. Niestety, nie można dzielić wyrazów w dowolnie wybranym miejscu: podział musi być zgodny z regułami ortografii. Potrzeba podziału zachodzi nie tylko przy pierwszym rozmieszczaniu tekstu: Najdrobniejsza korekta w rodzaju dopisania lub usunięcia słowa, związana z przepływem wątku, może pociągać za sobą wiele nowych podziałów słów w następujących wierszach. Pulpit wydawniczy ma więc na ogół wbudowaną bibliotekę typowych podziałów, a często dwie takie biblioteki: jedną standardową, a drugą — aplikacyjną, do której użytkownik może sam wpisywać terminy, nie występujące w słowniku głównym, takie, które należy podzielić w sposób niestandardowy lub których w danej publikacji w ogóle nie należy dzielić.

Jeżeli wątek nie mieści się w pierwszym bloku (sygnalizuje to kształt dolnego uchwytu), zacznę rozmieszczać bloki sąsiednie w kolejnych szpaltach, aż do wyczerpania wątku. Przypuśćmy, że w którymś miejscu, np. pośrodku szpalty, ma znaleźć się ilustracja. Uplasowawszy blok tekstu w tej szpalcie przesuwam myszką jego dolną krawędź w miejsce, w którym ma znaleźć się górna krawędź ilustracji. Następnie importuję gotową ilustrację z innego pliku dyskowego, który np. wybieram z menu. Mogę przy okazji zmienić oryginalną skalę ilustracji albo zmodyfikować ją tak, aby w polu widzenia zachować tylko istotne szczegóły. Poniżej ilustracji plasuję drugi w tej szpalcie blok tekstowy. Gdyby się okazało, że ilustracja ma szerokość np. półtorej szpalty, owinięcie jej tekstem zajęłoby mi kilkanaście sekund. Wystarczyłoby podzielić tekst na 5 oddzielnych bloków: 2 nad, 2 pod ilustracją, a 1 blok — obok ilustracji, w szpalcie, w której pozostało wolne miejsce.

Po rozmieszczeniu tekstu i ilustracji można zająć się ozdobnikami i efektami specjalnymi. Powszechnie stosowane są linie oddzielające poszczególne szpalty oraz uwypuklające tytuły i śródtytuły, a także ramki wokół ilustracji bądź wyodrębnionych partii tekstu. Można też spróbować „podłożyć" pod wybrane fragmenty publikacji płaszczyzny o różnym stopniu szarości lub deseniu wypełniającym albo poddać fragmenty tekstu inwersji (białe pismo na czarnym tle). Osobną sprawą jest tzw. kerning (dosuwanie), polegające na tym, że dla niektórych par liter lepszy efekt optyczny daje ich ciaśniejsze dosunięcie, niżby wynikało ze standardowej szerokości obu liter. Kerning można realizować ręcznie lub automatycznie, ale w praktyce daje on zauważalne efekty tylko dla czcionek o większych rozmiarach.

Klasyczne efekty specjalne polegają na łączeniu grafiki z tekstem. Przypuśćmy, że chcemy ilustrować publikację skanowanym zdjęciem lub rysunkiem. Chcielibyśmy jednak oznaczyć na tym rysunku szereg charakterystycznych punktów, do których występują odwołania w tekście. Przedsięwzięcie to zrealizujemy korzystając z koncepcji warstw. Zaczynamy od uplasowania rysunku na stronicy — będzie on stanowił najniższą warstwę. Następnie na rysunku, na odrębnej warstwie plasujemy 1 lub więcej bloków tekstowych, zawierających odpowiednie napisy. Napisy te możemy następnie precyzyjnie przesuwać i korygować ich położenie. Gdyby napisy na rysunku miały go zaciemniać, użyłbym innej metody, sporządzając poza obszarem rysunku opisy wybranych

szczegółów, a potem rysując cienką kreską linie odniesienia, łączące poszczególne napisy z odpowiednimi punktami rysunku.

Gdyby w trakcie pracy nad publikacją zabrakło mi pustych stronic, mógłbym dodać je w dowolnym miejscu publikacji, także pomiędzy stronicami już wypełnionymi. Mogę oczywiście także usuwać stronice puste lub wypełnione. Na koniec jeszcze krótki przegląd całości, kilka kosmetycznych poprawek i publikacja może być skierowana na drukarkę. Czas trwania wydruku może być różny, zależnie od użytej drukarki, ale to niewiele mnie już obchodzi, gdyż podczas druku mogę bez przeszkód pracować nad nową publikacją.

Od chwili rozpoczęcia prac nad moją 30-stronicową broszurą nie minęły 4 godziny. Prawda, że teksty i ilustracje miałem przygotowane już wcześniej, tak że przy ich rozmieszczeniu dokonywałem co najwyżej drobnych korekt. Musiałem tylko podpisać ilustracje i opisać dwie z nich kilkoma komentarzami z liniami odniesienia. Prawda, że pracowałem intensywnie, rezygnując z polskiego rytuału profesjonalnego, polegającego na popijaniu herbaty co 3 przyciśnięcia klawisza.

Istotą nowoczesnych pulpitów wydawniczych jest komfort obsługi, uzyskany przez maksymalne przybliżenie operacji na ekranie do podobnych operacji, wykonywanych tradycyjnie za pomocą kleju i nożyczek na biurku redaktora. Pozostańmy przy przykładzie PageMakera. Także i w komputerze dysponujemy szeregiem narzędzi, w które możemy „uzbrajać" myszkę zależnie od potrzeby, wybierając je z widocznego na ekranie przybornika. Jest w nim m.in. pisak tekstowy, linijka, kadrownica oraz wzorniki szeregu typowych figur, w szczególności okręgów, elips oraz prostokątnych ramek o ostrych lub zaokrąglonych rogach (promień zaokrąglenia rogu także można wybierać z menu osobno dla każdej ramki). Aby np. przesunąć ilustrację lub blok tekstu, należy wybrać z przybornika wskaźnik w formie strzałki, wskazać wnętrze obiektu, po czym nacisnąć przycisk myszki i nie zwalniając go po prostu przesunąć obiekt na ekranie. Ruch obiektu będzie widoczny tak samo, jak przy przesuwaniu kartki po biurku. Bardziej złożone czynności są wybierane z menu, przy czym większość z nich ma także odpowiedniki w postaci odpowiednich kombinacji klawiszy.

Komu można polecić technikę DTP? Sądzę, że praktycznie każdemu, kto para się, na różną skalę, komunikowaniem ludziom informacji lub idei za pomocą słowa pisanego, a dotychczas posługuje się w tym celu powielaczowymi odbitkami materiałów, przepisywanymi na maszynie mniej lub bardziej niechlujnie przez panią Kasię, która także wkleja w przygotowane „okienka" zdjęcia lub szkice. Oto garść przykładów: broszury i ulotki propagandowe, gazetki zakładowe, instrukcje użytkowania wyrobów przemysłowych, a także prace dyplomowe, opracowania naukowe, itd. W każdej z tych dziedzin kładzie się nacisk na inne atuty pulpitu wydawniczego, ale wszędzie pozwala ona na uzyskanie nowych jakości i(lub) istotne zmniejszenie nakładu pracy.

Technika DTP może zaspokoić potrzeby zwłaszcza osób ambitnych i wrażliwych estetycznie, które od początku do końca chciałyby zachować pełną kontrolę nad swoim dziełem, bez obawy, że na etapie „produkcji" jakaś dobra dusza coś spaskudzi. Jak zwykle kij ma jednak dwa końce: zwiększona kontrola oznacza zarazem zwiększoną odpowiedzialność za ostateczny efekt. Są, co prawda, w naszym życiu dziedziny, których powyższa zasada zdaje się nie dotyczyć, ale DTP do nich nie należy...



TYM RAZEM BEZ LUTOWNICY

TADEUSZ GOLONKA

Od ZX Spectrum do pakietu TOBOS-DYD

Od momentu rozpoczęcia sprzedaży mikrokomputera ZX Spectrum (1982 — Wielka Brytania, 1984 w Polsce) trwały próby stworzenia narzędzi pozwalających na efektywne pisanie programów.
Rozwiązania te obejmowały:
— tworzenie nowych interpreterów języka BASIC (Beta-BASIC, Mega BASIC),
— implementacje na ZX Spectrum translatorów języków wysokiego poziomu takich jak Pascal, C, Forth i inne,
— opracowanie kompilatorów standardowego języka BASIC.
Nowe interpretery znalazły ograniczone zastosowanie, gdyż mimo dodatkowych możliwości (np. graficznych) czas realizacji programu był jeszcze dłuższy. Również grono użytkowników translatorów innych języków jest niewielkie (jedynie Pascal w wersji proponowanej przez firmę Hisoft był i jest wykorzystywany praktycznie).
Najwięcej nadziei wiązano ze stworzeniem narzędzia pozwalającego na znaczne przyspieszenie pracy programów pisanych w standardowym języku BASIC. Do lata roku 1986 powstało wiele kompilatorów stało- i zmiennoprzecinkowych, spośród których jedynie kompilatory całkowitoliczbowe: IS 48k i MCODERII dają zadowalające rezultaty. Niestety nie potrafimy wskazać żadnego programu użytkowego, do kompilacji którego (ze względu na ograniczone tylko do obliczeń całkowitoliczbowych) można by kompilatory te wykorzystać. Kompilatory pełne: FP 48k, czy Blast ze względu na niską efektywność (1.1—3-krotne skrócenie czasu realizacji programu), posiadane ograniczenia (brak zgodności z interpreterem), niewygodę obsługi (Blast) również nie były tym czego oczekiwali użytkownicy mikrokomputera.



Pierwszym kompilatorem, który znalazł szerokie zastosowanie był opracowany w 1986 roku przez J. Borkowskiego i W. Skabę z Torunia TOBOS-FP. Niewielka liczba ograniczeń spowodowała, że większość programów napisanych uprzednio mogła działać w postaci skompilowanej.
Zalety tego kompilatora to:
1. Krótki czas realizacji programow (dzięki zastosowaniu nowej biblioteki procedur arytmetycznych).
2. Zgodność z interpreterem zapisanym w pamięci ROM.
3. Krótki czas kompilacji.
4. Niewielki obszar pamięci zajmowanej przez kompilator.

Kompilator ten był na tyle interesujący, że w odpowiedzi na postulaty użytkowników powstała nowa wersja, która wraz z zestawem dodatkowych procedur tworzy pakiet TOBOS-DYD.

Pakiet TOBOS-DYD zawiera:

— nową wersję kompilatora TOBOS-DYD (wer. 2.0),
— zestaw procedur narzędziowych,
— dodatkowe procedury graficzne (np. wypełnianie figury tłem, instrukcja CIRCLE wykonywana jest 5 razy szybciej),
— procedury współpracy z pamięcią zewnętrzną,
— dodatkowe możliwości trybu tekstowego (dowolna wielkość znaków, polskie litery).

Całość jest nierozdzielna i łącznie zajmuje 14796 bajtów pamięci powyżej adresu 50740.
Praca z pakietem pozwala na:
— przejście od pracy interpretacyjnej do kompilacyjnej (patrz pkt 4),
— kompilowanie dużych programów z wymazaniem zbioru źródłowego z pamięci,
— poprawianie programów z wykorzystaniem procedur narzędziowych:

* automatycznej numeracji linii (STEP od,co),
* przenumerowania linii programu (LINE start, skok),
* usuwania bloku linii (NOT od,do), a tym samym przenoszenie wybranych fragmentów programu jako procedur do innych programów,

— umieszczanie w programie napisów w języku polskim o dowolnej wielkości znaków,
— nagrywanie i wczytywanie danych przy współpracy z magnetofonem lub stacją dysków. Wprowadzono instrukcje LOAD, VERIFY oraz SAVE (tylko z opcją CODE lub SCREEN$), oraz nagrywanie i wczytywanie tablic,
— opuszczenie skompilowanego programu, realizację programu pod kontrolą interpretera języka BASIC i ponowne uruchomienie skompilowanego programu z zachowaniem wartości wszystkich zmiennych.

Praca z programem i pakietem TOBOS-DYD

Dużą zaletą opisywanego pakietu jest to, że napisane uprzednio programy będą działać znacznie szybciej. Pozwala to na „odświeżenie" wszystkich programów znajdujących się w bibliotece użytkownika ZX Spectrum.

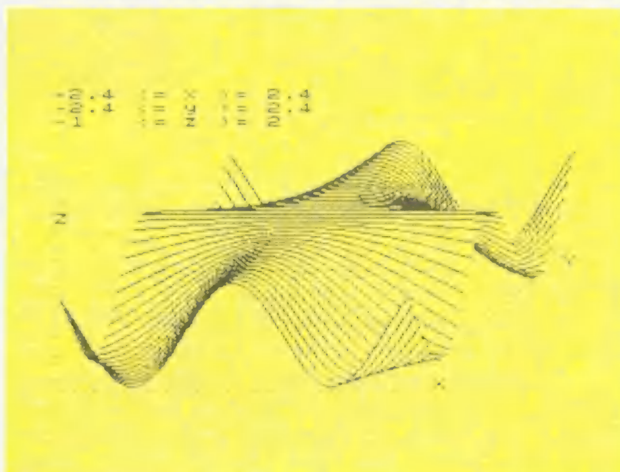
Jako przykład przedstawimy program „Foxes and Rabbits" z kasety Horizons znany wszystkim użytkownikom mikrokomputera. Po załadowaniu programu (w obecności pakietu) w linii 9400 instrukcję
LET w = USR 32000 zamieniamy na LET w = USR 51972, gdyż pakiet zawiera procedurę drukowania dużych napisów umieszczoną pod tym właśnie adresem. Następnie usuwamy zbędne instrukcje: LOAD, SAVE i CLEAR w lokalizacji czego pomaga nam próba kompilacji programu — znalezienie jednej z wymienionych instrukcji sygnalizowane jest jako błąd (usuwamy zbędne linie 8540, 9000, 9050). Kompilujemy program ponownie, a następnie uruchamiamy skompilowany program. Symulacja 30 miesięcy programu trwająca 58 sekund w przypadku pracy pod interpreterem zostanie skrócona do 12 sekund przy pracy z kompilatorem. Jeżeli dodatkowo z linii 4000 usuniemy instrukcję CIRCLE, to czas tego fragmentu symulacji wyniesie jedynie 6.5 sekundy.

Pisanie i uruchamianie nowych programów

Najwięcej czasu przy tworzeniu nowego programu zajmuje jego uruchamianie i testowanie poprawności pracy. Krótki czas kompilacji (ok. 50 wierszy/sekundę) w połączeniu z szybką pracą skompilowanego programu skraca czas lokalizacji i usuwania błędów. Możliwe jest zrezygnowanie z pracy pod interpreterem i uruchamianie programu tylko w postaci skompilowanej. Umieszczenie w programie jako pierwszej instrukcji:
1 IF 1/2-0.5 THEN RANDOMIZE USR 53100: RANDOMIZE USR 40000: STOP
zapewnia automatyczną kompilację i przejście do wykonywania skompilowanego programu po naciśnięciu RUN i ENTER. Pokażemy to na przykładzie programu rysującego wykresy funkcji dwóch zmiennych z = f x,y (wydruk 1).

```
 10 REM Sz. Slupik, T. Golonka
 20 IF 1/2-.5 THEN RANDOMIZE USR 53100: RANDOMIZE USR 40000: STOP
 30 DEF FN a(x,y)=COS (x*y): LET a$="COS (x*y)": LET zx=-2.4: LET sx=2.4:
    LET zy=-2.4: LET sy=2.4: LET zz=-1.0: LET sz=2.0: LET kr=2
 40 BORDER 0: PAPER 0: INK 7: CLS
 50 DIM z(255): DIM y(255): DIM x(42)
 60 FOR n=1 TO 255: LET y(n)=175: NEXT n
 70 CLS
 80 PRINT #0;"  f(x,y) = ";a$
 90 PRINT AT 0,0;
100 PRINT zx;TAB 4;" <= x >= ";sx
110 PRINT zy;TAB 4;" <= y >= ";sy
120 PRINT zz;TAB 4;" <= z >= ";sz
130 LET sx=(sx-zx)/170: LET sy=(sy-zy)/85: LET sz=90/(sz-zz)
140 FOR n=0 TO 170 STEP 5
150 PLOT n,0: REM n+85,85
160 NEXT n
170 FOR n=0 TO 90 STEP 5
180 PLOT 0,n
190 NEXT n
200 FOR n=0 TO 85 STEP 5
210 PLOT n+170,n
220 NEXT n
230 PRINT AT 21,23;"X": PRINT AT 12,31;"Y": PRINT AT 9,0;"Z"
240 FOR m=2 TO 85 STEP kr
250 LET y=m*sy+zy: LET u=0
260 FOR n=1 TO 170
270 LET i=n+m: LET x=n*sx+zx
280 LET z=(FN a(x,y)-zz)*sz+m
290 IF INKEY$<>"" THEN INK 7*(PEEK 22528<7): BORDER 7*(PEEK 22528<7):
    PAPER 7*(PEEK 22528<56): LET t=56*(PEEK 22528<7)+7*(PEEK 22528<56):
    FOR p=22528 TO 23295: POKE p,t: NEXT p
300 IF (z(i)>y(i) AND z<z(i) AND z>y(i)) OR z<m THEN LET u=0: GO TO 340
310 IF z>m+90 THEN LET u=0: GO TO 350
320 IF u THEN DRAW i-PEEK 23677,z-PEEK 23678: GO TO 340
330 LET u=1: PLOT i,z
340 IF z>z(i) THEN LET z(i)=z
350 IF z<y(i) THEN LET y(i)=z
360 NEXT n: NEXT m
370 STOP
```

Parametry określające zakres argumentów x,y, rozmiar funkcji na osi z i inne trzeba dobierać eksperymentalnie na podstawie tworzonego rysunku. Przykładowo program rysujący funkcję z = cos x*y (wydruk 2) wymaga każdorazowo 22 minut przy pracy z interpreterem oraz 1.5 minuty przy pracy z kompilatorem. Zakładając konieczność wykonania 10 prób oszczędności czasowe będą liczone w godzinach.



TOBOS-DYD a programowanie w asemblerze

W programie rysującym wykresy funkcji postaci z = F(x,y) zaistniała potrzeba umieszczenia segmentu zamieniającego kolory tła i pisaka po wciśnięciu dowolnego klawisza (poprawia to czytelność rysunku).

Rozwiązanie problemu wymaga dokonania zamiany całego obszaru atrybutów. Możemy tego dokonać pisząc odpowiedni program:
— w języku BASIC,
— w asemblerze,
— w języku BASIC, ale w obecności pakietu TOBOS-DYD.

Przykładowy program w języku BASIC:

```
IF INKEY$<>"" THEN INK 7*(PEEK 22528<>7): BORDER 7*(PEEK
22528=7): PAPER 7* PEEK(22528=56): LET t=56* (PEEK
22528=7)+7*(PEEK 22528=56): FOR p=22528 TO 23295: POKE p,t:
NEXT p
```

wymaga aż 14.0 sekund na wykonanie zadania. Pracując z pakietem TOBOS-DYD ta sama zamiana kolorów wykonana zostanie w czasie 0.5 sekundy (a więc 28-krotnie krótszym). Relokowalny program w asemblerze (wydruk nr 3) wykonuje się 0.03 sekundy. Mimo to rezultat działania programu skompilowanego można uznać za zadowalający. Tym samym przy pracy z pakietem TOBOS-DYD często stosowane odwoływanie się do pracochłonnych procedur asemblerowych nie jest konieczne.

```
210058        ld   hl,22528  ;poczatek atrybutow
110058        ld   de,23296  ;koniec atrybutow
7E       loop ld   a,(hl)
17            rla            ;wybor INK
17            rla
17            rla
E638          and  #38
47            ld   b,a       ;przechowanie INK
7E            ld   a,(hl)
1F            rra            ;wybor PAPER
1F            rra
1F            rra
E607          and  #7
4F            ld   c,a       ;przechowanie PAPER
7E            ld   a,(hl)
E6C0          and  #C0       ;zamiana PAPER i INK
B0            or   b
B1            or   c
77            ld   (hl),a    ;wpisanie nowego
23            inc  hl        ;atrybutu
7A            ld   a,d       ;czy koniec petli ?
BC            cp   h
20E7          jr   nz,loop
7B            ld   a,e
FD            cp   l
20E3          jr   nz,loop
C9            ret

    wydruk 3 - listing programu w asemblerze
```

TOBOS-DYD a programowanie w języku Pascal

Ciekawie przedstawia się porównianie pakietu TOBOS-DYD z translatorem Hisoft-Pascal, uważanym przez wielu za najefektywniejszy kompilator zmiennoprzecinkowy wykorzystywany na ZX Spectrum.

Okazuje się, że czasy realizacji programów wymagających obliczeń zmiennopozycyjnych są porównywalne. Tabela 1 zawiera czasy realizacji programu rysującego muchę z okładki czasopisma „Bajtek" (patrz „Komputer" nr 7/86 str. 21) oraz standardowego testu numerycznego (patrz „Informatyka" nr 6/85).

Wyniki programów testowych pozwalają na ocenę efektywności kodu wynikowego kompilatorów Pascala i BASIC-a, nie są natomiast podstawą do porównywania obu języków programowania.

Mimo iż autor wyżej stawia Pascal od BASIC-a, to w przypadku mikrokomputera ZX Spectrum uruchamianie programu w BASIC-u z pakietem TOBOS-DYD jest szybsze i efektywniejsze.

Praca pakietu TOBOS-DYD z przykładowym programem dydaktycznym

Jako przykład wykorzystania pakietu TOBOS-DYD do programów dydaktycznych w tabeli 2 zaprezentowano czasy realizacji siedmiu przekształceń geometrycznych tej samej figury złożonej z trzech okręgów i dwóch łamanych w programie „Przekształcenia

Tab. 1 — Rezultaty testów (czas dla języka Basic ZX Spectrum = 1.0)

	test 1	test 2	dokładność
BASIC	1.00	1.00	9E-09
TOBOS-DYD	0.09	0.04	1E-06
Pascal	0.11	0.07	3E-05

test 1 — „mucha"
test 2 — obliczenia numeryczne

Tab. 2 — Porównanie czasów realizacji wybranych przekształceń geometrycznych (czas realizacji z wykorzystaniem pakietu TOBOS-DYD = 1.0)

przekształcenie	TOBOS-DYD	BASIC
translacja	1.0	21.5
obrót	1.0	16.1
symetria osiowa	1.0	14.2
jednokładność	1.0	14.3
afiniczne	1.0	4.0
konhoidalne	1.0	19.3
powinowactwo	1.0	12.1

geometryczne" do nauczania geometrii w klasie I LO (autor programu: Eugeniusz Jakubas — Zamość). Program ten, bardzo interesujący od strony dydaktycznej, pozwala uczniom na eksperymentowanie z różnymi przekształceniami (izometryczne, afiniczne, konchoidalne a także definiowane dowolnymi wzorami) i w ten sposób poszukiwanie ich własności. Jako jeden przyjęto tym razem czas realizacji przekształcenia z wykorzystaniem pakietu TOBOS-DYD. Tak więc dane w tabeli 2 pokazują, ile razy dłużej wykonywane byłoby przekształcenie bez kompilatora. Warto podać, że zaprezentowane czasy oznaczają przykładowo dla obrotu bez pakietu TOBOS-DYD 58.1 sekundy i 3.6 sekundy z pakietem.

Zamiast zakończenia

Omawianie w 1988 roku programu czy pakietu przygotowanego dla ZX Spectrum wygląda archaicznie. Należy jednak zaznaczyć, że każdy program czy komputer ma swoje przeznaczenie i grono użytkowników.

ZX Spectrum i kompatybilne (firm Timex, Unipolbrit czy Elwro) to rzeczywistość szkolna czy klubów komputerowych. Do nich adresowany jest artykuł. Dobre i ciekawe programy można pisać na każdym komputerze i nie należy się tego wstydzić, jak w przypadku poważnego miesięcznika „Elektronizacja", który by ozdobić okładkę swojego czasopisma (nr 5—6/87) komputerem wykorzystywanym jako urządzenie pomiarowe zamieścił zdjęcie komputera kompatybilnego z IBM PC/XT. Fakt ten nie byłby godny uwagi, gdyby nie to, że do monitora podłączono (oczywiście zdjęcie tego nie pokazuje) ZX Spectrum z działającym programem.

Nie wierzycie? Sprawdźcie sami!

Redakcja wyraża podziękowanie współautorowi pakietu TOBOS-DYD, p. Wojciechowi Skabie, za udostępnienie egzemplarza testowego.



ROLAND WACŁAWEK

NEC HERCULES CONTRA... SideKick

Nec Hercules contra plures — tak brzmiało rzymskie przysłowie, które wkuwałem na lekcjach łaciny w ogólniaku. Czasy się zmieniają — dziś Hercules kojarzy się miłośnikom mikroelektroniki raczej z popularną kartą graficzną do IBM PC/AT niż z greckim herosem. Karta Hercules ma wiele mocnych punktów, ale i ją może niekiedy zmóc nagromadzenie przeciwności.

Przyczyną większości kłopotów z kartą Hercules jest omawiana już na łamach „InforMika" niemożność programowego ustalenia w prosty sposób bieżącego trybu pracy karty Hercules, co z kolei wynika z konsekwentnego ignorowania karty Hercules przez firmę IBM i związanego z tym braku wsparcia ze strony BIOS. Tak więc każdy program aplikacyjny we własnym zakresie inicjuje graficzny tryb pracy karty Hercules, wpisując dane wprost do jej rejestrów. Niezależnie od tego, czy karta pracuje w trybie tekstowym, czy graficznym, wywołanie funkcji usługowej nr 15 przerwania BIOS nr 16 nieodmiennie melduje tekstowy tryb pracy ekranu nr 7, przewidziany pierwotnie dla karty MGA.

Dopóki program aplikacyjny, używający kart Hercules w trybie graficznym, włada komputerem niepodzielnie, opisane fakty nie mają znaczenia. Co będzie jednak, jeżeli podczas eksploatacji programu zechcemy skorzystać z usług jakiegoś rezydującego narzędzia w rodzaju popularnego pakietu SideKick, np. w wersji 1.10?

Programy rezydujące po uruchomieniu pozostają w pamięci na stałe i mogą być uaktywniane np. określoną kombinacją klawiszy. W programie SideKick jest to ⟨Ctrl⟩+⟨Alt⟩. Powoduje to czasowe zawieszenie programu pierwszoplanowego. W chwili takiego wywołania SideKick wykonuje następujące czynności:

— Zapamiętuje w wydzielonym buforze zawartość tego fragmentu pamięci ekranu, który będzie mu potrzebny do własnych potrzeb.
— Odczytuje z BIOS i zapamiętuje bieżący tryb pracy ekranu.
— Włącza ustalony tryb tekstowy i wyświetla na ekranie swoje własne menu lub od razu okienka poszczególnych aplikacji.

Jeśli zechcemy opuścić SideKick lub podobny program rezydujący (w SideKicku służy do tego klawisz ⟨Esc⟩), SideKick odtworzy zarówno pierwotną zawartość pamięci ekranu, jak i tryb graficzny, zarejestrowany



w chwili wywołania. Chodzi o to, aby zawieszony program w żaden sposób nie odczuł, że w międzyczasie komputerem dysponował inny program. Ta szlachetna idea świetnie funkcjonuje z kartą CGA i EGA, ale nie daje się w pełni urzeczywistnić z kartą Hercules. Dlaczego?

Przypuśćmy, że wywołaliśmy SideKick w chwili, gdy ekran pracował w trybie graficznym. SideKick odczytał z BIOS bieżący tryb pracy ekranu, uzyskując kod 7, po czym włączył własny, tekstowy tryb pracy. Grafika znikła, na ekranie pojawiły się okienka SideKicka. Problem powstanie dopiero przy opuszczaniu SideKicka, gdy ten zechce odtworzyć pierwotny tryb, za który uważa tryb 7. Jest to tryb tekstowy... — i w takim trybie karta graficzna zostanie po powrocie do programu aplikacyjnego. Na ekranie uzyskamy więc „kaszę", a jedyną deską ratunku będzie opuszczenie „w ciemno" programu aplikacyjnego i jego ponowne uruchomienie. Nie zawsze się to udaje... Czy zatem będziemy musieli nieodwołalnie zrezygnować z SideKicka i podobnych narzędzi rezydujących przy pracy z tak wspaniałymi programami jak FRAMEWORK, MS-WORD czy AutoCAD tylko dlatego, że korzystają one z grafiki? Odpowiedź jest przecząca, ale zanim zabrzmi to wiarygodnie, trzeba się będzie trochę napracować.

Aby móc kontynuować dialog z zawieszonym programem pierwszoplanowym, musimy mieć możliwość ponownego włączenia trybu graficznego po opuszczeniu SideKicka. Samodzielny program uruchamiany z poziomu systemu operacyjnego nie wchodzi w grę, gdyż nie chcemy przecież przerywać, usuwać pierwszoplanowego. Musimy zatem odpowiedni program załadować już wcześniej — nawet jeszcze przed załadowaniem SideKicka — i pozostawić go w pamięci na stałe. Wywołanie tego programu może odbywać się podobnie jak SideKicka, tzn. zleceniem klawiszowym. Mamy więc wybijanie klina klinem...

Chcielibyśmy, aby przywrócenie grafiki odbywało się równocześnie z opuszczeniem SideKicka. Uczulimy więc nasz program na kombinację klawiszową ⟨Shift⟩+⟨Esc⟩. SideKick traktuje ją tak, jak sam klawisz ⟨Esc⟩ i także kończy aktywność. W ten sposób naciskając ⟨Esc⟩ wrócimy do trybu tekstowego, a ⟨Shift⟩+⟨Esc⟩ — do graficznego. Pozostał jeszcze jeden problem: karta Hercules ma 2 stronice pamięci graficznej, jedną pod adresem 0B000H (strona nr 0), a drugą pod adresem 0B800H (strona nr 1). Różne programy wykorzystują różne stronice graficzne. Którą z nich uaktywnić? Problem rozwiążemy w sposób najprostszy: kombinacja klawisza ⟨Esc⟩ z lewym klawiszem ⟨Shift⟩ uaktywni stronicę nr 0, kombinacja z prawym klawiszem ⟨Shift⟩ — stronicę nr 1. Jeżeli po powrocie do programu aplikacyjnego na ekranie ujrzymy „kaszę", będzie to zapewne wywołane błędnym wyborem stronicy graficznej. Należy wówczas spróbować użyć kombinacji z drugim klawiszem ⟨Shift⟩.

W chwili wywołania program zapamiętuje adres pierwotnej procedury przerwania nr 9, zastępuje go własnym wektorem wskazującym procedurę Obsługa, instaluje tę procedurę na stałe w pamięci, po czym kończy pracę. Po zainstalowaniu każde przerwanie nr 9, które jest wywoływane po każdym naciśnięciu, a także zwolnieniu przycisku klawiatury (omówiono to dokładnie w jednym z wcześniejszych numerów „InforMika") spowoduje wywołanie procedury: Obsługa.

Nasz program ma reagować wyłącznie na kombinacje z klawiszem ⟨Esc⟩. Pierwszą czynnością programu po przechowaniu rejestrów będzie odczytanie portu klawiatury w celu ustalenia kodu klawisza. 7 najmłodszych bitów reprezentuje numer klawisza; klawisz ⟨Esc⟩ nosi numer 1 (nie mylić z kodem ASCII znaku ⟨ESC⟩, który wynosi 27!). Najstarszy bit informuje, czy miało miejsce naciśnięcie klawisza (= 0), czy też jego zwolnienie (= 1). Gdyby na klawiaturze użyto innego przycisku, wówczas nastąpi skok wprost do pierwotnej procedury obsługi przerwania.

Nasz program będzie reagować na zwolnienie klawisza ⟨Esc⟩, dlatego testowany jest kod 10000001B = 81H. Dlaczego nie reagować na naciśnięcie? Powody są dwa. Pierwszy — to zapobieganie wielokrotnej inicjacji karty Hercules przy dłuższym przytrzymaniu klawisza ⟨Esc⟩. Drugi powód jest istotniejszy i wiąże się z działaniem SideKicka. Otóż po naciśnięciu klawisza ⟨Esc⟩, dopiero po obsłudze przerwanie SideKick pobiera kod znaku za pośrednictwem przerwania nr 16H i dopiero wtedy przywraca pierwotny tryb graficzny (w naszym przypadku zawsze włączy tryb tekstowy). Tak więc włączenie grafiki następowałoby za wcześnie i należałoby je powtórzyć przez ponowne użycie kombinacji ⟨Shift⟩+⟨Esc⟩ już po opuszczeniu SideKicka. Jeżeli jednak zareagujemy dopiero na zwolnienie klawisza ⟨Esc⟩, to wówczas SideKick zdąży już wcześniej zakończyć pracę i nie włączy nam ponownie trybu tekstowego.

W razie stwierdzenia zwolnienia klawisza ⟨Esc⟩ program obsługi bada bieżący stan obu klawiszy ⟨Shift⟩, testując odpowiednie bity flagowe w bajcie statusowym klawiatury, obsługiwanym przez BIOS. Gdyby któryś z tych klawiszy był wciśnięty, program sprawdzi, czy chodzi o klawisz lewy, po czym przechowa wynik tekstu na stosie i wywoła pierwotną procedurę obsługi. Sekwencja: PUSHF i CALL jest równoważna rozkazowi INT, którego tu jednak nie można użyć z powodu wcześniejszego „przegięcia" wektora przerwania nr 9. Po powrocie z pierwotnej procedury na podstawie wcześniejszego testu ładowany jest do rejestru AH kod sterujący, odpowiedni dla danej stronicy graficznej, a następnie zaczyna się zwykły proces programowania karty Hercules. Dokładniejsze dane na ten temat włącznie z opisem poszczególnych portów i bitów sterujących można znaleźć w dokumentacji karty.

Oto kompletny listing asemblerowy programu DLAILONY.COM, realizującego omówione powyżej funkcje. Nazwa programu wynika stąd, że powstał on w swoim czasie na życzenie mojej żony Ilony, przy-

zwyczajonej do korzystania z SideKicka w najdziwniejszych okolicznościach. Niechaj program ten posłuży jako inspiracja do własnych poszukiwań. Można np. zmieniać kombinacje klawiszy albo realizować czynności niekoniecznie związane z kartą Hercules.

```
;**********      D L A I L O N Y   wersja 1.00        **********
;********** Roland Waclawek,  Siemianowice Sl. 1987 **********

          Program    SEGMENT PARA
          ASSUME  CS:Program, DS:Program

          ORG      100H
Start:    JMP      Instaluj

Klaw_Adr: DW      0                ;Tu będzie przechowany pierwot-
Klaw_Segm:DW      0                ;ny  wektor  procedury  obsługi

Obsluga:  PUSH    AX
          PUSH    DS               ;przechowaj rejestry  na stosie
          XOR     AX, AX           ;wyzeruj  rejestr procesora  AX
          MOV     DS, AX           ;wyzeruj rejestr segmentowy  DS
          IN      AL, [060H]       ;Odczytaj do AL port klawiatury
          CMP     AL, 81H          ;Kod zwolnienia klawisza [Esc]?
          JNE     Ignoruj          ;Jeżeli nie, pomiń dalsze testy
          ASSUME DS:Dane_BIOS
          TEST BYTE PTR StatusKL,11B ;Czy naciśnięto [Shift]?
          JZ      Ignoruj            ;Jeżeli  nie,  to  skocz
          TEST BYTE PTR StatusKL,10B ;Czy  to  lewy  [Shift]?
          ASSUME DS:Program
          PUSHF                      ;Wywołaj pierwotną  pro-
          CALL  CS:DWORD PTR Klaw_adr;cedurę  obsługi  w BIOS
          PUSH  SI
          PUSH  DX                 ;przechowaj rejestry  na stosie
          PUSH  CX
          MOV   SI,OFFSET ParamHerc  ;Adres tabeli parametrów
          MOV   AH,2               ;Kod sterujący  dla  stronicy 1
          JNZ   LewyShift          ;Skacz, jeżeli był lewy [Shift]
          MOV   AH,82H             ;Kod sterujący  dla  stronicy 0
LewyShift:MOV   DX,03BFH           ;Adres  portu  konfiguracyjnego
          MOV   AL,11B             ;Kod  pełnej  konfiguracji  HGC
          OUT   DX,AL              ;Pisz kod konfiguracji do portu
          MOV   AL,AH              ;Kod sterujący trybem ekranowym
          MOV   DL,0B8H            ;DX:adres portu sterującego HGC
          OUT   DX,AL              ;Zapisz  kod  trybu graficznego
          MOV   CX,0               ;Zeruj licznik bajtów sekwencji
          MOV   DL,0B4H            ;DX :adres rejestru indeksowego
Petla:    MOV   AL,CL              ;Nr  kolejnego bajtu  sekwencji
          OUT   DX,AL              ;Wpisz  do rejestru indeksowego
          MOV   AL,CS:[SI]         ;Ładuj  kolejny bajt  sekwencji
          INC   SI                 ;Ustaw adres  następnego  bajtu
          INC   DX                 ;DX : 3B5H : adres portu danych
          OUT   DX,AL              ;Zapisz  kolejną daną  do portu
          DEC   DX                 ;Przywróć w DX adres portu ind.
          INC   CX                 ;Zwiększ o  1 licznik sekwencji
          CMP   CL,12              ;Czy to już ostatni,  12  bajt?
          JNZ   Petla              ;Jeśli nie, wyprowadzaj kolejne
          MOV   AL,AH              ;Kod  sterujący ponownie do  AL
          OR    AL,08H             ;Ustaw bit  widoczności  obrazu
          MOV   DL,0B8H            ;DX:adres portu sterującego HGC
          OUT   DX,AL              ;Zapisz  kod  trybu graficznego
          POP   CX                 ;Odtwórz  wykorzystane rejestry
          POP   DX
          POP   SI
          POP   DS
          POP   AX
          IRET                     ;Powrót z obsługi przerwania  9

Ignoruj:  POP   DS                 ;Odtwórz  wykorzystane rejestry
          POP   AX
          JMP   CS:DWORD PTR Klaw_adr ;Skok do pierw. pr. obs.

ParamHerc:DB    035H,02DH,02EH,07H,05BH,02H,057H,057H,2,3,0,0



Instaluj  PUSH  CS
          POP   DS                 ;Skopiuj  zawartość  CS  do  DS
          MOV   AH, 35H
          MOV   AL, 9H
          INT   21H                ;Odczytaj do ES:BX  wektor nr 9
          MOV   WORD PTR Klaw_Adr, BX     ;Przechowaj  offset
          MOV   WORD PTR Klaw_Segm,ES     ;Przechowaj segment
          MOV   DX,OFFSET Obsluga
          MOV   AH,25H
          MOV   AL, 9H
          INT   21H                ;Wpisz nowy wektor przerw. nr 9
          MOV   DX,OFFSET Instaluj
          INT   27H                ;Uczyń   program   rezydującym

Program   ENDS

Dane_BIOS SEGMENT AT 0
          ORG      417H            ;bajt stanu Klaw.-parametr BIOS
StatusKL  LABEL    BYTE            ;bit 0: prawy Shift, bit 1:lewy
                                   ;bit 2: Ctrl, bit 3 : Alt
Dane_BIOS ENDS

          END      Start
```

Aby uzyskać działający program, należy zapisać tekst źródłowy w pliku np. DLAILONY.ASM i zainicjować asemblację:

MASM DLAILONY, , , ,

a następnie konsolidację:

LINK DLAILONY, , , ,

Ostatnią czynnością jest transformacja z formatu EXE na COM:

EXE2BIN DLAILONY.EXE DLAILONY.COM

Program najlepiej wywoływać zleceniem DLAILONY zaraz po uruchomieniu systemu operacyjnego, jeszcze przed ładowaniem innych programów rezydujących, chociaż nie jest to krytyczne. Kombinacja ⟨Lewy Shift⟩+⟨Esc⟩ włącza tryb graficzny ze stronicą nr 0, kombinacja ⟨Prawy Shift⟩+⟨Esc⟩ — ze stronicą nr 1. Jak widać, ubocznym zastosowaniem programu może być przełączanie stronic graficznych. W niektórych programach, np. w systemie TURBO-Graphics, karta Hercules jest inicjowana w trybie nie 348, lecz 352 linii.

Aby uzyskać ten sam efekt w powyższym programie, należy zastąpić w sekwencji ParamHerc obydwa sąsiednie bajty o wartości 57H przez 58H.

Dla tych, którym program DLAILONY może się przydać, ale nie mają ochoty na zabawę z Asemblerem, przygotowałem jak zwykle program generacyjny — tym razem w TURBO-Pascalu. Program wystarczy wprowadzić i uruchomić. Jeżeli przy wpisywaniu danych nie popełniono omyłki (program sprawdza to obliczając sumę kontrolną), w katalogu roboczym zostanie utworzony plik DLAILONY.COM. W razie wykrycia błędu zabrzmi sygnał akustyczny i pojawi się odpowiedni meldunek.

```
PROGRAM Laduj_DLAILONY;
{Roland Waclawek, Siemianowice Sl. 11 lipca 1988}
CONST Ost_bajt = $BF;
      Kod_masz: ARRAY[0..Ost_bajt] OF byte =
        ( $EB, $6F, $90, $00, $00, $00, $00, $50,
          $1E, $33, $C0, $8E, $D8, $E4, $60, $3C,
          $81, $75, $4B, $F6, $06, $17, $04, $03,
          $74, $44, $F6, $06, $17, $04, $01, $9C,
          $2E, $FF, $1E, $03, $01, $56, $52, $51,
          $BE, $65, $01, $B4, $82, $75, $02, $B4,
          $02, $BA, $BF, $03, $E0, $03, $EE, $8A,
          $C4, $B2, $B8, $EE, $B9, $00, $00, $B2,
          $B4, $8A, $C1, $EE, $2E, $8A, $04, $46,
          $42, $EE, $4A, $41, $80, $F9, $0C, $75,
          $F0, $8A, $C4, $0C, $08, $B2, $B8, $EE,
          $59, $5A, $5E, $1F, $58, $CF, $1F, $58,
          $2E, $FF, $2E, $03, $01, $35, $2D, $2E,
          $07, $5B, $02, $57, $57, $02, $03, $00,
          $00, $0E, $1F, $B4, $35, $B0, $09, $CD,
          $21, $89, $1E, $03, $01, $8C, $06, $05,
          $01, $BA, $07, $01, $B4, $25, $B0, $09,
          $CD, $21, $BE, $A0, $01, $AC, $34, $F7,
          $33, $DB, $B4, $0E, $CD, $10, $80, $3C,
          $00, $75, $F2, $BA, $71, $01, $CD, $27,
          $B3, $BB, $B6, $BE, $BB, $B8, $B9, $AE,
          $D7, $D7, $A5, $98, $9B, $96, $99, $93,
          $D7, $A0, $96, $94, $9B, $96, $80, $92,
          $9C, $D7, $CF, $C0, $FA, $FD, $00, $00);
          Suma: integer= 20832;
VAR i    : integer;
   Plik : FILE OF byte;
BEGIN Assign(Plik, 'DLAILONY.COM');
      ReWrite(Plik);
      FOR i:= 0 TO Ost_bajt DO
         BEGIN
            Write(Plik,  Kod_masz[i]);
            Suma:= Suma- Kod_masz[i]
         END;
      IF Suma:0 THEN
                BEGIN Close(Plik);
                  Writeln('Program zapisany!')
                END
              ELSE Writeln(#7'Bledne dane!!!')
END.
```

REGENERATOR KSZTAŁTU SYGNAŁU MAGNETOFONOWEGO

PAWEŁ KOTKOWSKI

DARIUSZ A. PRZYGODA

Opisane poniżej urządzenie pełni funkcję regeneratora kształtu sygnału wyjściowego magnetofonu. W zamyśle przeznaczone ono było dla użytkowników zestawu składającego się z mikrokomputera ZX Spectrum i magnetofonu, jednak praktyka wykazała, że może być ono przydatne także i użytkownikom innych typów komputerów.

Urządzenie może pracować jako:

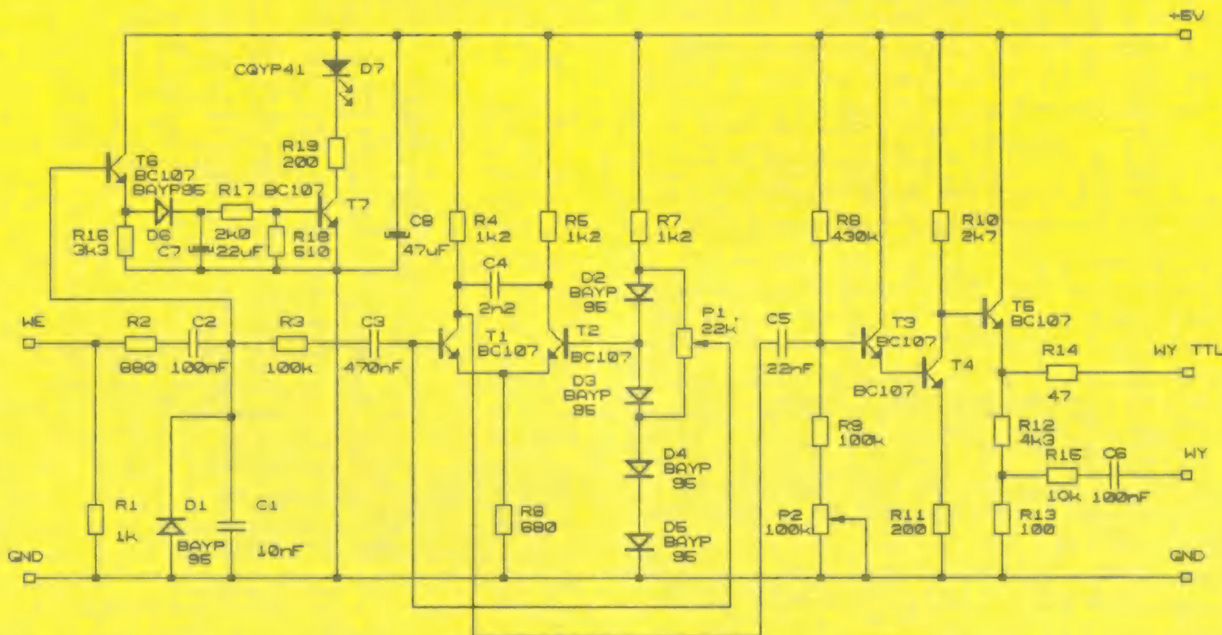
1. Blok pośredniczący pomiędzy dwoma magnetofonami przy kopiowaniu programów bez użycia komputera (metodą „taśma—taśma");
2. Blok regeneratora sygnału pomiędzy magnetofonem i komputerem. Drugi tryb pracy umożliwia niekiedy odzyskanie zawartości pliku, który (ze względu na niezgodności sprzętu, na którym plik był nagrywany i sprzętu służącego do jego odtwarzania) nie daje się wczytać do komputera.

Z układowego punktu widzenia urządzenie stanowi dwa wzmacniacze połączone kaskadowo, wyposażone w zestawy filtrów i regulatory punktów pracy. Zaprojektowanie go w technice dyskretnej (jedynie z użyciem tranzystorów) umożliwiło dostosowanie układu do zasilania pojedynczym napięciem +5V, co ułatwia eksploatację.

Schemat ideowy urządzenia przedstawiony jest na rys. 1. Można w nim wyróżnić bloki funkcjonalne wyszczególnione i omówione poniżej.

Blok filtrów wejściowych (R1—R3, C1—C3, D1). Zadaniem tego bloku jest filtracja szumów i zakłóceń sygnału wejściowego.

Wzmacniacz I (T1, T2, D2, D3—D5, R4—R7, C4, P1). Jest to wzmacniacz różnicowy z wyjściem asymetrycznym. Diody D3—D5 służą do stabilizacji napięć polaryzujących tranzystory T1 i T2, potencjometrem P1 kompensuje się napięcie niezrównoważenia spowodowane asymetrią tranzystorów (jego niezerowa wartość może powodować zniekształcenia symetrii od-



Rys 1. Schemat ideowy układu.

tworzonego sygnału). Kondensator C4 zapobiega wzbudzaniu się układu.

Wzmacniacz II (T3—T5, R8—R15, C5—C6, P2). Jest to wzmacniacz w układzie Darlingtona z wyjściem o małej impedancji wyjściowej. Potencjometr P1 służy do ustawienia punktu pracy wzmacniacza. Dużą obciążalność wyjścia zapewnia stopień z tranzystorem T5 pracujący w układzie wspólnego kolektora. Właściwe poziomy napięć na wyjściach zapewnia dzielnik R12/R13, a odporność na zwarcia wyjścia WY TTL — rezystor R14.

Wskaźnik poziomu napięcia wejściowego (T6—T7, D6—D7, R16—R19, C7). Jest to układ umożliwiający ustawienie optymalnego poziomu napięcia wejściowego. Stopień w układzie wspólnego kolektora zrealizowany na tranzystorze T6 zapewnia dużą impedancję wejściową układu, elementy D6, C7, R17 i R18 tworzą prostownik z filtracją i podziałem napięcia wyjściowego, T7 jest wzmacniaczem końcowym sterującym diodę elektroluminescencyjną D7.

Sposób montażu urządzenia nie jest krytyczny i, jako taki, jest pozostawiony do uznania osób powielających układ.

Do uruchomienia układu niezbędny jest miernik uniwersalny i oscyloskop. Uruchamianie sprowadza się do:

1. Ustawienia za pomocą potencjometru P2 napięcia na wyjściu WY TTL równego progowi przełączania układów TTL, czyli ok. 1.3V (regulacji dokonuje się przy zwartym wejściu WE);
2. Ustawienia punktu pracy wzmacniacza różnicowego. W tym celu należy posłużyć się nagraniem fali prostokątnej (najlepiej wykonać je zgrywając na magnetofon zawartość fragmentu obszaru pamięci zapełnionego bajtami zerowymi). Sygnał z magnetofonu doprowadzamy do wejścia WE (poziom głośności ustawiamy tak, aby dioda D7 lekko się świeciła) i, obserwując na oscyloskopie przebieg na wyjściu WY TTL, ustawiamy jego symetrię potencjometrem P1.

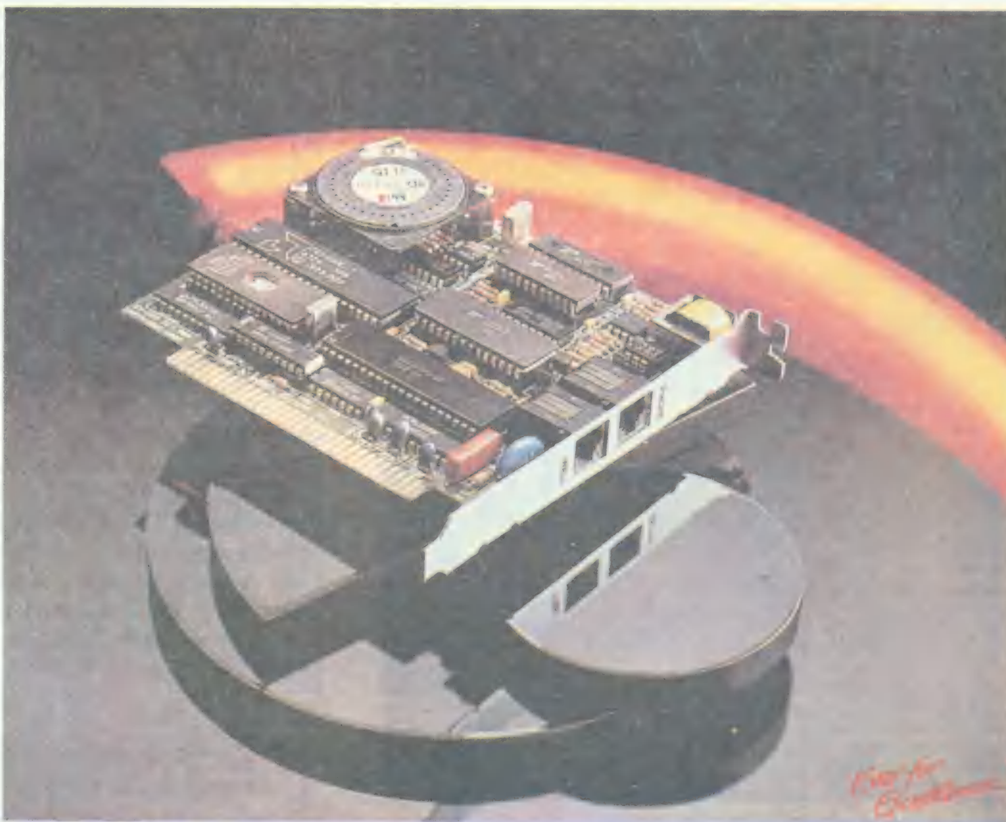
Po dokonaniu powyższych regulacji układ jest gotowy do pracy. W przypadku użytkowania go jako kopiarki sygnał do nagrywania pobieramy z wyjścia WY układu, w przypadku używania go jako regeneratora wejście komputera dołączamy do wyjścia WY TTL układu.

Wadą opisanego urządzenia jest to, że przypadkowe sygnały wejściowe o wystarczająco dużej amplitudzie przenoszą się na wyjście w postaci fali prostokątnej. Jest to niestety cena, którą trzeba zapłacić za prostotę urządzenia (należy tu zauważyć, że opisany układ stanowi adaptowany blok układu formującego „inteligentnego" układu kopiującego wyposażonego w oddzielny system mikroprocesorowy, który zapewniał filtrację sygnałów pasożytniczych w sposób programowy).

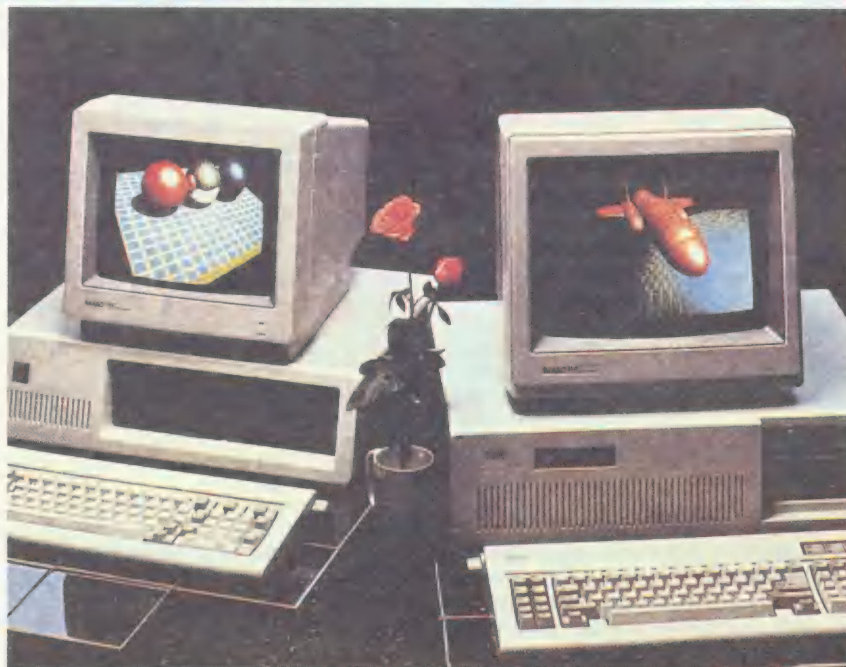
WYKAZ ELEMENTÓW

Tranzystory:

T1—T7	BC 107 (lub podobne),	
Diody:		
D1—D6	BAYP 95 (lub podobne),	
D7	CQYP 41 (lub inna elektroluminescencyjna).	
Rezystory (MŁT 0.125 W):		
R1	1k	
R2, R6	680	
R3	100k	
R4, R5, R7	1k2	
R8	430k	
R9	100k	
R10	2k7	
R11, R19	200	
R12	4k3	
R13	100	
R14	47	
R15	10k	
R16	3k3	
R17	2k0	
R18	510	
Kondensatory:		
C1	10 nF	ceramiczny
C2, C6	100 nF	,,
C3	470 nF	,,
C4	2,2 nF	,,
C5	22 nF	,,
C7	22 μF/16V	elektrolityczny
C8	47 μF/16V	,,
Potencjometry montażowe (np. TVP110):		
P1	22k	
P2	100k	



Modem jest to urządzenie sprzęgające komputer z linią telefoniczną, umożliwiające za jej pośrednictwem wymianę informacji pomiędzy nawet bardzo odległymi maszynami. Zdjęcie przedstawia modem firmy Evercom. Został on skonstruowany jako „krótka" karta do komputera IBM PC.



DARIUSZ ADAM PRZYGODA

KATALOG ELEMENTÓW NA TWARDYM DYSKU

Pośród dostępnego obecnie oprogramowania wspomagającego pracę inżyniera posługującego się przy pracy komputerem IBM PC można znaleźć wiele programów do zastosowania w elektronice, jak np. służący (między innymi) do sporządzania schematów OrCAD lub wspomagający projektowanie obwodów drukowanych SMARTWORK. Oba te programy znajdują zastosowanie w końcowej fazie realizacji gotowego projektu układu elektronicznego. Innym narzędziem wspomagającym, znajdującym zastosowanie przy wstępnych pracach projektowych, jest opisany poniżej pakiet programowy D.A.T.A. (CONDUSTRIE A.G.). Stanowi on specjalizowaną bazę danych, zawierającą dane katalogowe pamięci półprzewodnikowych. Pakiet przeznaczony jest do instalacji na komputerach zgodnych z IBM PC (XT lub AT) pracujących pod kontrolą systemu MS DOS. Do swojej prawidłowej pracy wymaga minimum 512 KB pamięci RAM i obecności dysku sztywnego.



Fot. 1. Główne menu programu

Pakiet składa się z dwóch programów będących oddzielnymi katalogami pamięci ROM oraz RAM. Według danych producenta zawarte są w nim dane katalogowe ponad 10 000 pamięciowych obwodów scalonych.

Producent dostarcza pakiet w postaci dziesięciu dyskietek z dołączonymi instrukcjami (dla każdego programu oddzielnie). Instrukcje, chociaż niezbyt elegancko wydane, są jednak bardzo starannie opracowane i zawierają dokładny opis instalacji i obsługi pakietu.

Same programy dostarczone są na dziesięciu dyskietkach $5^1/_4''$ w postaci skondensowanej. Procedura instalacyjna umieszcza je na dysku twardym, gdzie zajmują łącznie ok. 7 MB. Proces instalacji polega na umieszczeniu plików na dysku sztywnym i wstępnym uporządkowaniu tablicy indeksów (dokładniej będzie to opisane później) i jest długotrwały (ok. 25 min. dla jednego programu dla komputera XT turbo).

Same dyskietki nie są zabezpieczone przed kopiowaniem, a nawet instrukcja zaleca wręcz wykonanie kopii

roboczej do instalacji. Nie oznacza to jednak, że pakiet stanowi oprogramowanie typu public domain; dwie strony instrukcji zajmują notki o prawach autorskich i trybie postępowania w celu uzyskania od autorów licencji.

Obsługa programu jest prosta i sprowadza się do wyboru jednej z podanych na ekranie opcji (za pomocą cyfry lub rozjaśnionego paska naprowadzanego na opcję klawiszami kursora). Istnieje także możliwość wyświetlenia na ekranie „bryków" pomocnych w rozpraszaniu wątpliwości. Oprawa graficzna programu jest skąpa, ale elegancka.

Pakiet składa się wprawdzie z dwóch programów, jednakże struktura ich jest jednakowa, a drobne różnice sprowadzają się do występowania (lub nie) pewnych parametrów układów scalonych charakterystycznych dla rodzaju danego elementu (np. czas impulsu zapisu dla pamięci RAM lub wartość napięcia programującego dla pamięci EPROM).

Po wywołaniu program zgłasza się winietą producenta, a następnie na ekranie pojawia się główne menu (fot. 1), zawierające pięć opcji:

1. wyszukiwanie obwodu scalonego według typu elementu,
2. wyszukiwanie obwodu scalonego według wersji elementu,
3. selekcja obwodów scalonych według zadanych parametrów,
4. rekonfiguracja,
5. powrót do systemu

oraz ukrytą opcję pomocy (help) uruchamianą klawiszem ⟨F1⟩ (jest ona dostępna z każdego poziomu programu i zawiera wyszczególnienie klawiszy niezbędnych do sterowania programem).

Poniżej zawarte jest krótkie omówienie możliwości oferowanych przez poszczególne opcje.

Wyszukiwanie obwodu scalonego według typu elementu daje nam możliwość dokonania przeglądu wszystkich elementów jednego typu, bez względu na ich producentów (np. MCM2114P45 i MN2114-3 są to oznaczenia takich samych pamięci wytwarzanych przez różnych producentów i różniących się parametrami; „rdzeniem" nazwy stanowiącym typ elementu jest liczba 2114).

W przypadku nieznalezienia wyspecyfikowanego typu w zasobach program daje nam możliwość ponownej próby z innym numerem określającym typ kostki, możliwość bezpośredniego przejścia do opcji drugiej głównego menu lub też możliwość powrotu do głównego menu (rezygnacji).

W przypadku sukcesu program informuje nas o ilości znalezionych elementów (rozróżniając wersje elementu wytwarzane przez jednego producenta) oraz umożliwia m.in. wydrukowanie na drukarce listy znalezionych elementów, ponowną próbę z innym numerem lub wyświetlenie na ekranie danych elementu wybranego spośród wymienionych w liście. Ilustruje to fot. 2.

Opcja druga, czyli wyszukiwanie obwodu scalonego według wersji elementu, różni się od opcji pierwszej jedynie tym, że zadaną daną jest w tym wypadku dokładne oznaczenie kostki (zawierające wszystkie przedrostki i końcówki dodawane przez wytwórcę do liczby oznaczającej typ). Po znalezieniu wybranego układu program wyświetla na ekranie dane elementu (fot. 3).



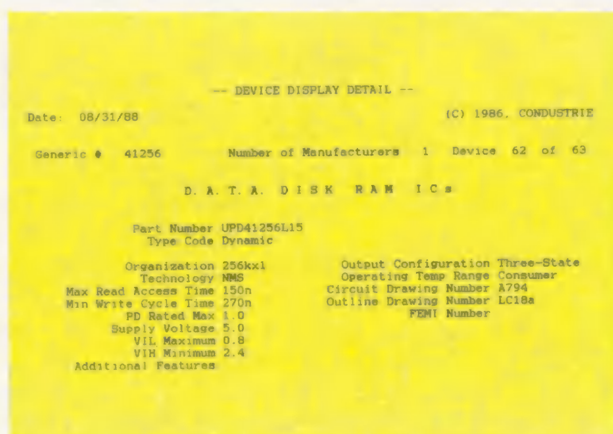
Fot. 2. Widok ekranu po znalezieniu szukanego elementu

W skład danych elementu podawanych przez program wchodzą:

— rodzaj pamięci (tzn. ROM, EPROM, RAM statyczna, RAM dynamiczna itp.),
— organizacja pamięci,
— technologia wykonania układu,
— dane elektryczne (napięcie(a) zasilające, poziomy napięć wejściowych i wyjściowych, napięcie programowania dla pamięci EPROM, moc rozpraszana itp.),
— dane dynamiczne (czas dostępu, czas trwania impulsu zapisu itp.),
— charakterystyka we/wy (rodzaj wyjść, ilość wejść zezwalających),
— zakres temperatury pracy,
— numery specyfikacji rozkładu wyprowadzeń i rodziny odpowiedników funkcjonalnych.

Ostatnie dane oznaczają numer rysunku z wyprowadzeniami elementu i indeks grupy układów scalonych stanowiących odpowiedniki funkcjonalne. Dane te zawarte są w katalogu dołączanym do pakietu przez firmę (ale na specjalne życzenie i za oddzielną opłatą).


Fot. 3. Sposób wyświetlenia parametrów wybranego elementu



Rys. 1. Wydruk parametrów wybranego elementu

Program zaznacza także typ wyświetlanej danej (graniczna, charakterystyczna) za pomocą znaków specjalnych (!, ?, #); listę tych znaków można przywołać na ekran naciskając klawisz ⟨F2⟩.

Z tego poziomu programu dostępne są dodatkowe opcje, jak np. natychmiastowe wyświetlenie parametrów innego elementu tego samego typu lub wyświetlenie danych o producencie kostki (wraz z adresem, numerem telefonu i teleksu). Ta ostatnia opcja wydaje się być mało przydatna w naszych warunkach.

Możliwe jest także skopiowanie danych na drukarkę (rys. 1).

Bardzo przydatna jest trzecia opcja głównego menu. Umożliwia ona selekcję elementów według określonych parametrów, jak np. rodzaju elementu (np. RAM statyczna), organizacji pamięci, technologii wykonania, czasu dostępu, zakresu temperatur pracy.

Po dokonaniu selekcji przez program możliwości obróbki danych są takie same jak w pierwszej i drugiej opcji.



Fot. 4. Wybór warunków selekcji parametrycznej elementów (uaktywniona opcja „rodzaj elementu")

Opcja czwarta daje nam następujące możliwości:

— wybór współpracy z monitorem (monochromatyczny lub kolorowy),
— uporządkowanie tablicy indeksów,
— modyfikacja danych,
— usunięcie z pamięci informacji o wytwórcach.

Istnienie drugiej z tych możliwości spowodowane jest sposobem optymalizacji czasu dostępu do zasobów. Konieczność jawnego jej wyboru istnieje w zasadzie w dwóch sytuacjach: podczas pierwszego uruchomienia



po instalacji oraz w wypadku uruchomienia programu po uprzednim jego „gwałtownym" przerwaniu (np. przez RESET); program może nie zdążyć pozamykać otwartych plików, co spowoduje błędne działanie przy ponownym uruchomieniu.

Czwarta z możliwości jest szczególnie przydatna w naszych warunkach — nieużywane praktycznie dane wytwórców obwodów scalonych zajmują sporo miejsca w pamięci!

Szybkość wyszukiwania danych przez program jest zadowalająca, czas selekcji określonego elementu wynosi średnio kilkanaście sekund (zależy to od egzemplarza komputera).

Do wad programu należy zaliczyć duże wymagania dotyczące zajętości pamięci na dysku sztywnym oraz brak topologii wyprowadzeń w danych katalogowych. Do zalet — występowanie takich parametrów, jak czasy dostępu i zapisu poszczególnych wersji pamięci. Duże możliwości kryje też w sobie możliwość selekcji parametrycznej elementów.

Załączona do instrukcji ulotka informuje, że zarejestrowani posiadacze programów mogą liczyć (po wykupieniu abonamentu) na dostarczanie przez firmę uaktualnień danych zawartych w programie. Uaktualnione dane przesyłane są klientowi co kwartał w postaci jednej lub dwóch dyskietek zawierających pliki katalogowe i program dołączający je do zasobów pakietu.



DARIUSZ ADAM PRZYGODA

INSTALACJA PAMIĘCI REPROGRAMOWALNEJ EPROM W KOMPUTERZE ZX SPECTRUM

Popularny mikrokomputer ZX Spectrum (w wersji 16 KB lub 48 KB) posiada 16 KB pamięci stałej ROM. Zawarty jest w niej program spełniający funkcje systemu operacyjnego i interpretera języka BASIC.

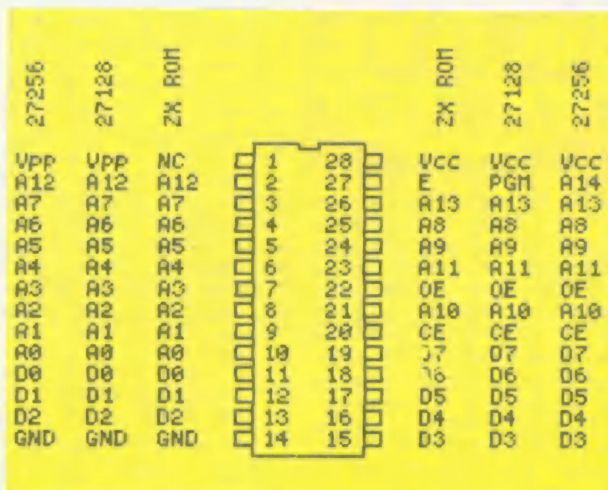
Poniższy artykuł stanowi opis sprzętowych przeróbek niezbędnych do zastąpienia pamięci stałej ROM pamięcią reprogramowalną EPROM o zmodyfikowanej zawartości.

Celem wprowadzania modyfikacji do programu systemowego komputera może być między innymi usunięcie błędów, będących wynikiem pośpiechu, w jakim program ten był tworzony przez programistów Sinclair Research Ltd.

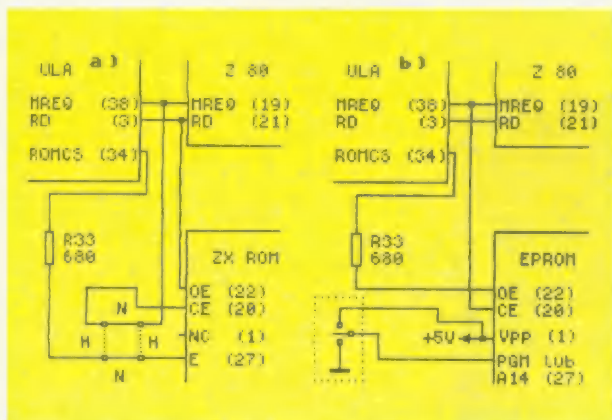
Dokładne informacje o błędach systemu i sposobie ich poprawiania zawierają m.in. pozycje „The Complete Spectrum ROM Disassembly" (dostępna na giełdach w postaci kserokopii) i broszurka Andrzeja Kadlofa „Tajniki ZX Spectrum" wydana w ramach serii wydawniczej KOMPUTER ABC.

Celem zmiany oprogramowania pamięci stałej komputera może być także umieszczenie w jej nie używanych obszarach własnych programów (np. obsługi urządzeń peryferyjnych).

Pamięcią EPROM odpowiadającą pod względem pojemności i organizacji wewnętrznej pamięci stałej ZX Spectrum jest układ 27128 (produkowany m.in. przez firmy Intel, OKI, NEC). Drobne różnice wyprowadzeń



Rys. 1. Rozkład wyprowadzeń układów pamięci stałych ZX ROM, 27128 i 27256



Rys. 2. Schemat połączenia wejść wewnętrznych dekoderów pamięci stałych w mikrokomputerze ZX Spectrum: a) dla pamięci stałej ZX ROM, b) dla pamięci reprogramowalnych 27128 i 27256

obu układów powodują, że wymagane jest dokonanie niewielkich zmian połączeń obwodu drukowanego komputera.

Przed dokonaniem zmiany trzeba jednak pamiętać, że istnieją programy wymagające do prawidłowej pracy niezmodyfikowanego oprogramowania systemowego. Programy te sprawdzają sumę kontrolną zawartości pamięci ROM (zabezpieczenie przed przerwaniem pracy przy pomocy NMI) lub też wykorzystują pusty obszar pamięci jako tablicę adresów przerwań lub wzorzec graficzny. Niezgodność zawartości pamięci stałej z oryginalną zawartością powoduje w takim wypadku nieprawidłową pracę programu lub zawieszenie się systemu.

Uniezależnić się od tego można zastępując ZX ROM pamięcią EPROM o dwukrotnie większej pojemności (typ 27256), w której zawarte będą oba programy: oryginalny systemowy i zmodyfikowany, dołączonej tak, że mikroprocesor będzie „widział" naraz tylko jedną z nich. Zewnętrzny przełącznik umożliwia wybór połówki pamięci w zależności od aktualnych potrzeb.

Analiza wyprowadzeń trzech wymienionych wyżej typów pamięci (patrz rys. 1) wykazuje, że zamiana jest równie łatwa dla pamięci 27128 jak i 27256.

Schemat połączeń ZX ROM wymagających modyfikacji przedstawia rys. 2a. Obecność zwor tłumaczy się tym, że producenci pamięci (firmy NEC i Hitachi) stosowali różne rozkłady wyprowadzeń dekoderów wewnętrznych układów. Schemat połączeń po modyfikacji przedstawia rys. 2b. Wyjaśnienia wymaga obwód przełącznika dołączonego do wyprowadzenia (27) układu pamięci EPROM. W przypadku układu 27256 na wyprowadzenie to podaje się najstarszy bit adresu (A14). Podając na nie stały poziom napięcia (niski albo wysoki) uaktywniamy „widzianą" przez mikroprocesor połówkę pamięci. Zaletą tego rozwiązania jest jego prostota, a wadą to, że wybór trybu pracy musi być dokonany przed włączeniem komputera do sieci (przełączenie w trakcie pracy może spowodować utratę kontroli nad wykonywanym programem). W przypadku pamięci 27128 dla prawidłowej pracy układu na wyprowadzenie to musi być podany stan wysoki, co realizujemy łącząc je galwanicznie z wyprowadzeniem (28) (+5 V) i usuwając przełącznik z obwodu.

Na zakończenie należy dodać, że wykonanie przeróbki wymaga — mimo jej prostoty — dużego stopnia zaawansowania, gdyż najczęściej ZX ROM jest wlutowany w obwód drukowany komputera. Nieumiejętne wylutowywanie obwodu zakończyć się może uszkodzeniem zarówno delikatnych ścieżek drukowanych, jak i sąsiednich elementów. Podkreślić też należy konieczność starannego dokonania przeróbki, gdyż ewentualne mikrozwarcia spowodować mogą uszkodzenia, których usunięcie będzie bardzo kosztowne.



Kasowalny dysk optyczny

Znana z produkcji materiałów dla fotografów firma Kodak opracowała dysk optyczny o pojemności 50 MB i średnicy 3.5", o wysokości równej połowie standardowej. Zastosowano termomagnetyczną zasadę zapisu danych, co pozwala na ich kasowanie i modyfikacje. Wbudowano układy detekcji i korekcji błędów, interfejs SCSI o przepustowości 1 MB/s. Średni czas dostępu do danych wynosi 70 ms.

Szwajcarska precyzja...

...jest tak znana, że aż dziw, iż dotąd nie słyszeliśmy o komputerach z tego kraju. Na wiosnę firma CSPI zademonstrowała procesor wektorowy, który można podłączyć do stosowanej w mini-komputerach magistrali VMEbus. Możliwe jest połączenie do czterech procesorów o wydajności 32 MFLOPS każdy. Procesory są wyposażone w 128 lub 512 KB szybkiej pamięci statycznej oraz kanały DMA. Pobór mocy wynosi 25 watów — dziesięć lat temu superkomputery o tej samej wydajności potrzebowały kilkudziesięciu kilowatów. Producent zapewnia 30 000 godzin bezawaryjnej pracy.

Skaner dla IBM PC

PC Scan 1000 holenderskiej firmy Dest może odczytać stronę tekstu formatu A-3 w ciągu 20 sekund. Rozdzielczość wynosi 0.08 mm, rozpoznawanych jest 16 stopni szarości. Skaner wyposażony w oprogramowanie firmy Dest jest w stanie współpracować z wieloma popularnymi edytorami tekstu, programami DTP (Desktop Publishing) i programami komunikacyjnymi. Wymiary urządzenia wynoszą 55 × 37 × 10 cm.

Klawiatura do wszystkiego...

...czyli Cherry 2000 przeznaczona dla IBM PC/XT/AT. Posiada 122 klawisze, w tym 24 klawisze funkcyjne. Dodatkowo została wyposażona w czytnik kart magnetycznych (wsuwanych w szczelinę z boku klawiatury) oraz wyświetlacz LCD mieszczący dwie linie tekstu. Do klawiatury można podłączyć mysz oraz czytnik kodów paskowych. Producent, zachodnioniemiecka firma Cherry Mikroschalter, opracował własny system operacyjny zarządzający zasobami klawiatury — CKOS.

Kolejna karta do IBM PC

Jest nią PC4000 firmy Silicon Composers ze słonecznej Kalifornii. Sercem karty jest mikroprocesor Novix NC4016 o architekturze RISC, mieści się na niej 512 KB pamięci, z której 16 KB jest dostępnych w przestrzeni adresowej IBM PC — obszar ten służy jako bufor do przekazywania danych. Całe oprogramowanie karty, także system operacyjny PCX, jest ładowane z dyskietki przez komputer. IBM może pracować równolegle z kartą lub tylko służyć jako stacja dostarczająca danych. Karta o długości $^{3}/_{4}$ długości karty standardowej oferuje moc 4—8 MIPS (zależnie od wykonywanych zadań), można też podłączyć się do sześciu kart uzyskując moc obliczeniową 42 MIPS.

Opracował **Adam Nowicki**

FACEBENDER

— elektroniczny karykaturzysta

Susan Brennan jest pracowniczką laboratoriów Hewlett-Packarda w Palo Alto w Kalifornii. W wolnych chwilach stworzyła program automatycznie sporządzający karykatury. Facebender został napisany na mikrokomputerze HP, jednak sądzę, że Czytelnicy bez trudu przeniosą jego ideę na dowolny komputer.

Rozpoczynając pracę z programem należy wprowadzić rysunek twarzy za pomocą pióra świetlnego lub myszy. Można również użyć digitizera do odczytania informacji ze zdjęcia. Wprowadzane są dane o 186 punktach charakterystycznych twarzy, jak kąciki warg czy czubek nosa. Program pyta o położenie kolejnych punktów — należy wtedy nasunąć kursor na stosowne miejsce i wcisnąć klawisz ENTER. Jeśli pojawiająca się na ekranie — linia po linii — twarz odpowiada wizerunkowi z fotografii to znaczy, że wprowadzanie danych przebiega prawidłowo. Każdemu punktowi zostają przyporządkowane dwa wymiary w osiach współrzędnych X i Y. Następnie wymiary części twarzy są zwiększane lub zmniejszane w stopniu, w jakim się różnią od przechowywanych w pamięci komputera rozmiarów przeciętnych.

W pamięci maszyny znajdują się cztery tablice o 186 wierszach i dwóch kolumnach. W tablicy face są przechowywane informacje o twarzy karykaturowanej osoby, w tablicy norm — o przeciętnej. Wyniki obliczeń umieszczane są w tablicy bend, natomiast w disp — współrzędne punktów ekranu w postaci zrozumiałej dla sterownika graficznego (najczęściej stosowany sposób przechowywania rysunków w pamięci komputera opisałem w „Komputerze", w odcinku 1/88 KMK). Przed zapisaniem danych w tablicy face należy je przetworzyć — zdjęcie osoby karykaturowanej może się różnić skalą od rysunku wzorcowego. Program zakłada, iż odległość między źrenicami oczu ludzi dorosłych jest jednakowa, zawsze przypisując źrenicom współrzędne (135,145) i (190,145).

Brudnopis artykułu pokazałem przyjacielowi z Akademii Medycznej, który się oburzył — rozstaw oczu jest cechą indywidualną każdego człowieka. Nie jest nawet proporcjonalny do wielkości głowy. Myślę jednak, że autorce programu rozrywkowego możemy wybaczyć pewne uproszczenia.

Po uwzględnieniu różnic skali niezgodności współrzędnych z tablic face i norm wynikają już tylko z charakterystycznych cech twarzy. Są one uwydatniane za pomocą procedury:

```
procedure przesadz;
    var i:integer;
    for i: = 3 to 186 do begin;
    bend (i,1) : = face (i,1) + fx (face (i,1)
            — norm (i,1)) ;
    bend (i,2) : = face (i,2) + fx (face (i,2)
            — norm (i,2)) ;
    end;
end;
```

Punkty o numerach 1 i 2 — źrenice oczu — jako jedyne pozostaną na swoich miejscach. Współczynnik f, nazywany Współczynnikiem Przesady, określa stopień zniekształcenia wizerunku. Jego ustalenie należy do użytkownika programu.

Punkty zostały uporządkowane w ten sposób, że można wykreślić twarz łącząc liniami punkty o kolejnych numerach. Jednak nie wszystkie pary punktów mają być połączone. W 37-elementowej tablicy features są przechowywane dane o ilościach punktów składających się na poszczególne części twarzy. Program wykreśla tyle linii, ile wynosi odpowiedni element tej tablicy, po czym przerywa łączenie punktów i wznawia od punktu należącego do następnego fragmentu twarzy.

Facebender oferuje kilka trybów pracy. Punkty mogą być zaznaczane kropkami lub małymi plamkami, jak na rysunku po lewej. Linie łączące są, zależnie od życzenia, wykreślane jako proste odcinki lub krzywe. Rysowanie krzywych trwa znacznie dłużej — ich wypukłość i promień krzywizny zależą od położenia. Daje to lepsze odwzorowanie zaokrąglonych kształtów twarzy ludzkiej, ale wydłuża czas potrzebny na obliczenia.

Program skrywa w sobie wiele niespodzianek. Potrafi sympatyczny uśmiech z fotografii zmienić w upiorny grymas. Modele są „podobne do siebie", gdy zdjęcie robiono en face — niewielkie nawet obrócenie głowy Facebender interpretuje jako różnicę w stosunku do twarzy przeciętnej i bezlitośnie wyolbrzymia domniemane niezgodności.

Wygląd twarzy przeciętnej stworzono metodą prób i błędów. Powstał wzorzec twarzy męskiej, żeńskiej i schematyczny rysunek twarzy nieokreślonej płci. Ciekawe efekty daje użycie zdjęcia znanej osoby, na przykład skrzyżowanie fizjonomii koleżanki i Tiny Turner.

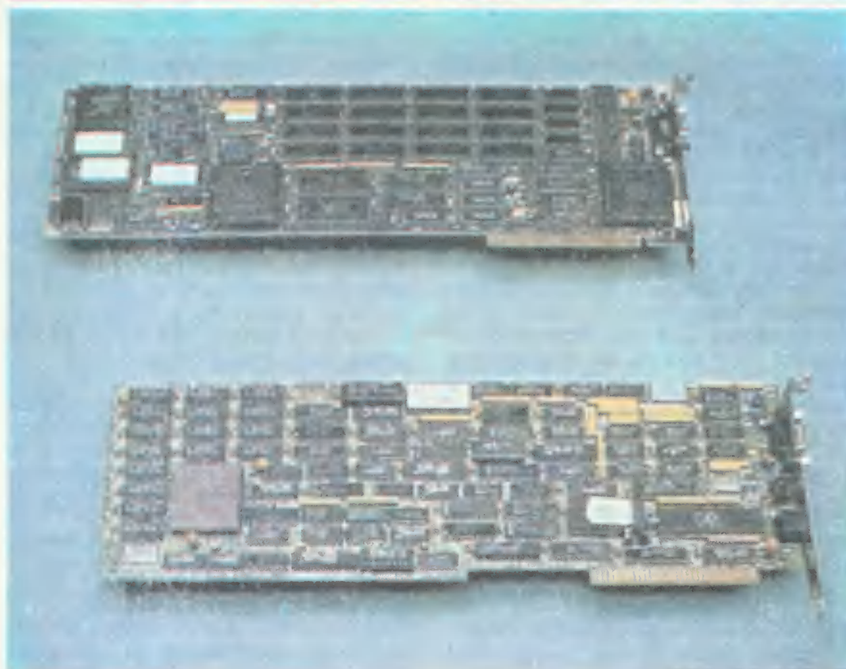
Pani Brennan przeprowadziła doświadczenia ze studentami wydziału psychologii uniwersytetu w Stanford. Pokazywała im zdjęcia sławnych ludzi i ich karykatury wykonane za pomocą programu Facebender. Okazało się, że osobę z karykatury można rozpoznać niemal dwukrotnie szybciej, niż na zdjęciu. Łatwiej uchwycić charakterystyczne cechy twarzy stanowiące o tożsamości. Nie opuszcza mnie wizja komisariatów wyposażonych w komputery, na ekranach których świadkowie rozpoznają karykatury bandziorów...

Poniżej przedstawiamy trzy twarze tej samej osoby. Po lewej rysunek będący zdigitalizowaną wersją oryginalnego zdjęcia. W środku i po prawej — karykatury wykonane przy użyciu różnych współczynników f.

wg „Scientific American" 5/86
opracował **Adam Nowicki**







ANDRZEJ KURYŁOWICZ

Karty sterowników graficznych do komputerów standardu IBM PC/XT/AT I PS/2

Sterownik graficzny należy do podstawowego wyposażenia każdego komputera standardu IBM PC (i nie tylko) stanowiąc interfejs pomiędzy jednostką centralną a monitorem. Informacja przeznaczona dla użytkownika mikrokomputera przetwarzana jest przez sterownik na odpowiedni sygnał wizyjny, analogowy lub cyfrowy, który steruje dołączonym monitorem.

Najczęściej sterownik graficzny jest wykonany w postaci tzw. karty i zainstalowany w jednym z gniazd rozszerzających (złączu krawędziowym) magistrali komputera umieszczonym na jego płycie głównej. W niektórych komputerach (np. Amstrad 1512 czy modelach serii IBM PS/2) sterownik znajduje się bezpośrednio na płycie głównej. Użytkownik skazany jest wówczas na używanie sterownika danego typu; włożenie dodatkowej karty innego sterownika może (choć nie musi) powodować konflikt ze sterownikiem już zainstalowanym, objawiający się np. zawieszaniem się systemu lub pojawieniem się „sieczki" na ekranie przy uruchamianiu niektórych programów. Tego typu rozwiązanie jest jednak wygodne, eliminując potrzebę stosowania dodatkowych kart rozszerzających możliwości graficzne, gdy zainstalowany sterownik ma charakter uniwersalny, a więc jest w stanie emulować wiele trybów graficznych i tekstowych (patrz Tabela 1). Niektóre firmy tajwańskie oferują obecnie płyty do XT ze sterownikiem pracującym w trybach Hercules i CGA oraz do AT (i 386) ze sterownikiem odpowiadającym funkcjonalnie karcie Super-VGA-Hi-Res firmy Genoa Systems i bazującym na specjalizowanym graficznym układzie scalonym tej firmy (tryby Hercules, EGA, CGA, VGA, 800 × 600 itd.).

W konstrukcjach kart graficznych zaznaczyła się wyraźna ewolucja, objawiająca się z jednej strony zwiększaniem skali integracji stosowanych układów scalonych i zmniejszaniem ich ilości (a co za tym idzie — rozmiarów kart), a z drugiej strony poszerzaniem ich możliwości funkcjonalnych. Coraz częściej do budowy kart stosuje się oprócz układów o średniej skali integracji układy o dużej (LSI) i bardzo dużej (VLSI) skali integracji. Rezultatem jest zwiększanie niezawodności, przy jednoczesnej obniżce kosztów produkcji, co pociąga za sobą obniżkę cen. Dobrą ilustracją tego zjawiska jest spadek, w przeciągu zaledwie 1,5 roku, cen kart EGA z poziomu 300—500 $ do poziomu 150—300 $, przy jednoczesnym znacznym poszerzeniu ich możliwości (procesu tego nie powstrzymał nawet 4—5-krotny wzrost cen pamięci dynamicznych RAM).

Na rynku dostępne są pojedyncze układy scalone zdolne do emulacji wszystkich podstawowych trybów łącznie z nowym standardem VGA, produkcji takich firm jak Chips & Technologies, Tseng Labs (seria ET3000), Paradise, Genoa Systems, Cirrus Logic, Everex, Trident

MicroSystems (TVGA). Aby stać się samodzielnym producentem karty graficznej o dużej rozdzielczości wystarczy „obudować" taki układ kilkoma pomocniczymi układami scalonymi i dołączyć generator kwarcowy.

W zamieszczonej w artykule tabeli zestawiono niektóre istotne parametry znanych standardów sterowników graficznych. W dniu dzisiejszym znaczenie niektórych z tych konstrukcji silnie się zdewaluowało; inne (np. MCGA) z góry skazane były na niepowodzenie; są też takie (np. VGA), przed którymi przyszłość stoi dopiero otworem.

Sterowniki CGA i MDA, pierwsze z wprowadzonych przez firmę IBM, mają obecnie już tylko znaczenie historyczne.

Karta MDA została całkowicie wyparta przez kartę Hercules (HGC) o tej samej rozdzielczości tekstowej, ale wyposażonej dodatkowo w monochromatyczny tryb graficzny o wysokiej rozdzielczości (720 × 348 punktów). Tekst wyświetlany przez CGA nie odznacza się dobrą wyrazistością, i z tego względu karta ta wykorzystywana była przede wszystkim w kolorowym trybie graficznym; jak na obecne możliwości techniczne liczba dostępnych kolorów jest niewielka, nawet w porównaniu z niektórymi komputerami domowymi. Praktycznie wszystkie programy o charakterze półprofesjonalnym lub w pełni profesjonalnym wyposażone są w procedury obsługi (ang. *drivers*) co najmniej 2—3 sterowników graficznych; najczęściej są to standardy Hercules,CGA i EGA. Użytkownik dysponujący monitorem monochromatycznym korzysta oczywiście z tego pierwszego ze względu na wyrazistą grafikę i tekst; użytkownik, któremu zależy na grafice kolorowej posłuży się najczęściej monitorem kolorowym o podwyższonej rozdzielczości (tzw. monitorem EGA), niewiele droższym (o 100—120 $) od zwykłego monitora RGB. Gdzie tu miejsce na kartę CGA? Nic więc dziwnego, że przestała ona odgrywać jakąkolwiek rolę w zastosowaniach profesjonalnych (z założenia takie jest przeznaczenie mikrokomputerów standardu IBM PC) i przydaje się wyłącznie do uruchamiania nielicznych gier, w tym popularnych w Polsce Diggera, Tetris, Gato czy Flight Simulator. Zresztą EGA może pracować również w trybie CGA; istnieją też programowe emulatory karty CGA na kartę Hercules (np. program Color). Do współpracy z CGA można stosować zwykły monitor monochromatyczny — koniecznie jednak typu „dual mode", a więc dający się synchronizować zarówno sygnałami o częstotliwości 15,75 kHz (CGA), jak i 18,43 kHz (MDA i Hercules).

Dominującym standardem jest obecnie karta sterownika Hercules, szczególnie wygodna przy pracy z edytorami tekstowymi. Karta ta wykonywana była oryginalnie jako karta o pełnej długości; od pewnego czasu jej funkcjonalne kopie wykonuje się jako tzw. karty krótkie. Niektórzy producenci uzupełniają ją o sprzętową emulację karty CGA — zapewne w tym celu, aby użytkownik mógł połączyć pracę na komputerze z przyjemną zabawą.

Karty sterownika EGA w swojej oryginalnej wersji nigdy nie zyskały popularności, i od pewnego czasu całkowicie zaprzestano ich produkcji. Najpierw barierę psychologiczno-ekonomiczną stanowiła ich relatywnie wysoka cena (4—8 razy wyższa niż kart Hercules i CGA, nie mówiąc już o cenie monitora), potem, gdy ceny zaczęły gwałtownie spadać pojawiły się znacznie ulepszone karty EGA (enhanced EGA): HEGA (EGA + Hercules), Super EGA (HEGA + 5—20 trybów dodatkowych, zależnie od producenta). Od ok. 2 lat trwa



Obraz uzyskiwany na monitorze Ultrasync (rozdzielczość 1050 × 770 punktów)

wyścig producentów oferujących na 1 karcie (najczęściej krótkiej) coraz to nowe tryby pracy: graficzne, m.in. 640 × 480, 640 × 528, 640 × 400 (CGA double scan), 752 × 410, 800 × 560, 800 × 600, 1024 × 768 (16 kolorów z palety 64 możliwych barw) oraz tekstowe: 80 × 28, 80 × 32, 80 × 44, 132 × 25, 132 × 44, 132 × 60 itd. (132 kolumny tekstu — szczególnie przydatne przy korzystaniu z programów arkuszy kalkulacyjnych w rodzaju Lotus 1-2-3 czy zastosowaniach typu desktop publishing). Wśród najbardziej znanych warto wymienić karty EGA firm Genoa Systems (Super EGA, Super EGA HiRes — patrz zdjęcie), ATI Technologies (EGA Wonder, EGA Wonder 800), Paradise Systems (AutoSwitch EGA480), Quadram (Quad EGA), Orchid Technologies (Orchid EGA, Turbo EGA), Bocca Research (MultiEGA), Video 7 (VEGA, VEGA DeLuxe). Karta EVA480 firmy Tseng Labs wyróżnia się efektowną

Karta graficzna Super EGA HiRes firmy Genoa (rozdzielczość 800 × 600 punktów)

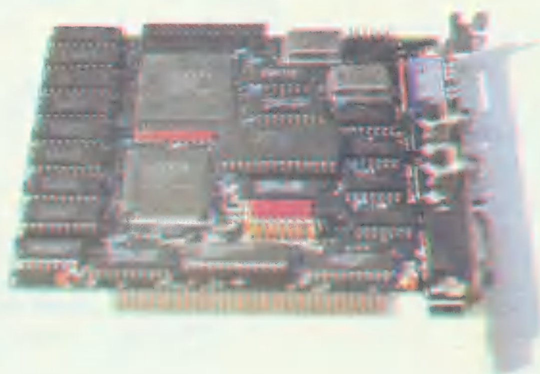
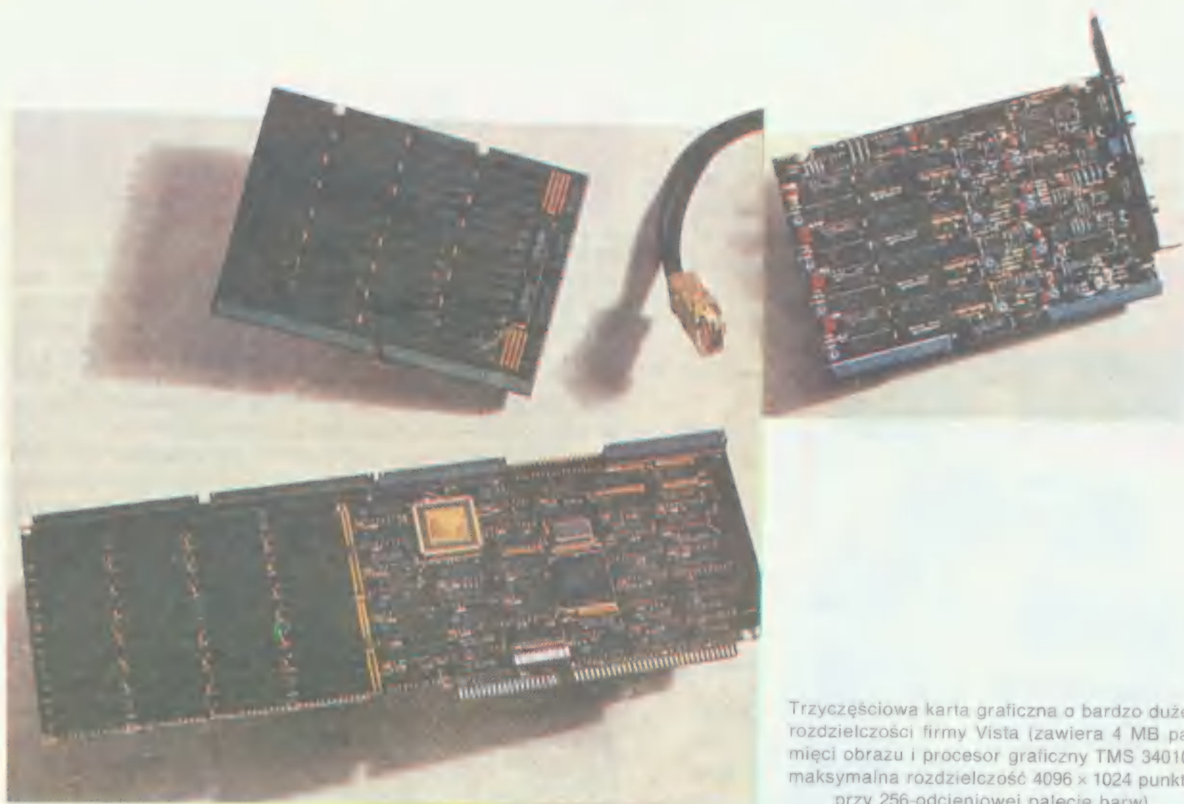


TABELA 1: PODSTAWOWE TYPY STEROWNIKÓW GRAFICZNYCH

1	2	3	4	5
CGA (Color Graphics Adapter) 16 kB	320x200 4 kolory z 16 z 2 palet 640x200 mono	80x25 16 kol. z 16 8x8	1) kolorowy z wej. cyfr. RGB lub anal. (composite video) 2) odbiornik TVC (NTSC) + modulator RF 3) monochromat. z wej. TTL lub analog. (composite video) 15.75 kHz	1) 9-stykowe RGBI TTL 2) analogowe zespol. sygnalu wizyjnego (composite video)
MDA (Monochro Display Adapter) 4 kB	-	80x25 mono 8x14	monochromatyczny z wej. TTL 18.43 kHz	9-stykowe TTL
HGC (Hercules Graphics Card) 64 kB	720x348 mono	80x25 mono 9x14	monochromatyczny z wej. TTL 18.43 kHz	9-stykowe TTL
EGA (Enhanced Graphics Adapter) 64/256 kB	640x350 16 kol. z 64 *emuluje CGA*	80x25 16 kol z 64 8x14	kolorowy o podwyższonej rozdzielczości z wej. cyfr. RGB (monitoror EGA) 21.85 kHz	9-stykowe RGBI TTL
PGC (Professional Graphics Controller)	640x480 256 kol. z 4096	80x25 256 kol z 4096	multisynchroniczny 30.4 kHz	?
MCGA (Multi-Color Graphics Array) 64 kB	320x200 256 kol. z 262144 *emuluje CGA*	80x25 16 kol. z 262144 8x16	1) kolorowy z wej. analog. RGB. 2) monochromatyczny z wej. analog. 31.5 kHz	15-stykowe analogowe
VGA (Video Graphics Array) 256 kB	640x480 16 kol. z 262144 *emuluje MCGA, EGA*	80x25 16 kol, z 262144 9x16	1) kolorowy o wysokiej rozdzielczości z wej. analog. RGB (monitor VGA) 2) monochromatyczny o podwyższonej rozdzielczości z wej. analog. (monitor VGA) 3) multisynchroniczny (monitor VGA) 31.5 kHz	15-stykowe analogowe

1. Typ sterownika / rozmiar pamięci RAM na karcie
2. Maksymalna rozdzielczość w trybie graficznym + liczba dostępnych kolorów
3. Maksymalna rozdzielczość w trybie tekstowym + liczba dostępnych kolorów / rozmiar matrycy znaku
4. Wymagany monitor (częstotliwość synchronizacji poziomej)
5. Typ wyjścia monitorowego



Trzyczęściowa karta graficzna o bardzo dużej rozdzielczości firmy Vista (zawiera 4 MB pamięci obrazu i procesor graficzny TMS 34010, maksymalna rozdzielczość 4096 × 1024 punkty przy 256-odcieniowej palecie barw)

funkcją „zoom", czyli powiększania wybranego wycinka ekranu (2-, 4- lub 8-krotnie).

Wspólną cechą wszystkich produkowanych obecnie kart EGA (ulepszonych — bo innych się już nie wytwarza!) jest, oprócz 256 KB pamięci obrazu (pamięć RAM na karcie), tryb grafiki kolorowej 640 × 480 punktów — obowiązujący, choć oficjalnie nie uznawany standard. Trybu tego nie należy mylić z trybem PGC czy VGA, pomimo tej samej rozdzielczości. Inne są bowiem liczby dopuszczalnych kolorów, a także np. lokalizacja BIOS-ów EGA i VGA w przestrzeni adresowej. (W kartach emulujących oba mody EGA i VGA stosuje się przełączanie pomiędzy EGA BIOS a VGA BIOS).

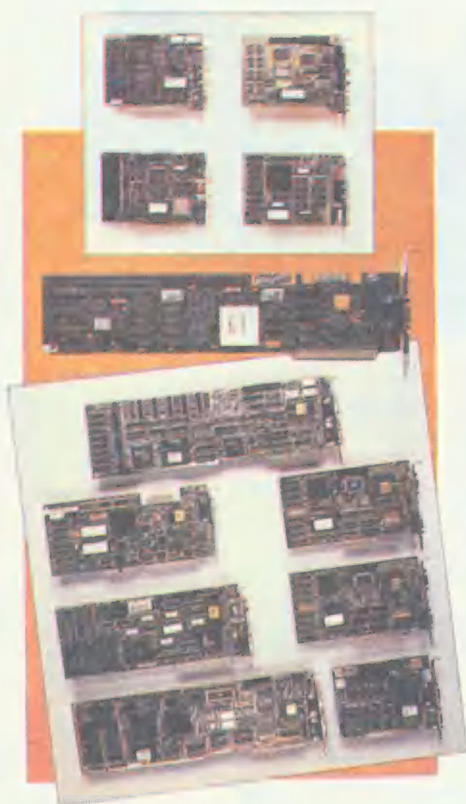
Próbę wylansowania nowego standardu podjęła firma Hercules Computer Technology proponując kartę Hercules InColor, tj. zwykłą kartę Hercules wyposażoną dodatkowo w grafikę, możliwość pracy z tekstem w 16 kolorach (z palety 64) i możliwość definiowania zestawu własnych znaków oprócz wzorcowych 256 znaków ASCII zapisanych w pamięci ROM (tzw. RamFont: max. 3072 znaki w 16 kolorach). Karta ta nie zyskała przewidywanej przez producenta popularności, głównie ze względu na wysoką cenę (400—500 $), silną konkurencję i fakt, że wielu użytkowników przyzwyczaiło się do innego standardu. Brak poparcia ze strony firm software'owych przeważył szalę na jej niekorzyść.

Karta Professional Graphics Controller firmy IBM, o cenie powyżej 1000 $ z założenia przeznaczona była do zastosowań typu CAD/CAM. Wysoka cena, a zarazem niezbyt rewelacyjna rozdzielczość grafiki (jak na aktualne możliwości technologiczne) sprawiły, że o sterowniku tym słuch zupełnie zaginął. Do prac projektowych wykorzystuje się w chwili obecnej sterowniki o znacznie wyższych: rozdzielczości (rzędu 1024 × 1024 lub większej) i szybkości działania, wyposażonych w sprzętową realizację szeregu funkcji uprzednio spełnianych przez oprogramowanie (np. rysowania podstawowych figur geometrycznych, wypełniania fragmentów obrazu, zmiany skali czy prostej animacji).

Jedną z najsłabszych stron komputerów IBM PC był przyjęty przez firmę standard graficzno-tekstowy CGA/MGA. Spodziewano się, że jego następcą będzie EGA — stało się jednak inaczej. W nowej rodzinie komputerów serii PS/2 firma IBM wprowadziła dwa nowe sterowniki graficzne: MCGA i VGA (zainstalowane na płycie głównej).

MCGA to po prostu ulepszony sterownik CGA. Poprawiono czytelność znaków w trybie tekstowym zwiększając rozmiar matrycy znaku do 8 × 16 punktów, zwiększono liczbę dostępnych kolorów. Pomimo tych zmian nie ma raczej wątpliwości, że MCGA to propozycja chybiona. Wymagania użytkowników mikrokomputerów sięgają obecnie znacznie wyżej. Dowodzi tego fakt, że żaden z działających na rynku komputerowym producentów nie podjął się wytwarzania kopii tego sterownika, zaś firma IBM zastosowała go tylko w modelu PS/2 30, którego produkcji już zaniechano.

Oczekiwania wszystkich spełnia natomiast sterownik VGA — naprawdę wyśmienity standard graficzny, szkoda, że wprowadzony tak późno. Charakteryzuje go nie tylko doskonała grafika kolorowa, ale także rozdzielczość tekstowa wyższa niż karty Hercules. Doceniły to natychmiast firmy software'owe: nowe wersje popularnych programów (arkuszy kalkulacyjnych, baz danych, kompilatorów) zostały wyposażone w procedury umożliwiające współpracę z VGA (firma IBM, przekonana o swojej dominacji takich procedur nie dostarczyła). Równocześnie rozpoczął się wyścig producentów kart graficznych, starających się jak najszybciej zaoferować karty sterowników VGA dla ogromnej rzeszy użytkowników mikrokomputerów PC/XT/AT/386 (komputerów nie zawierających magistrali Micro Channel). Należą do



Karty EGA (wersja ulepszona) i VGA

nich m.in. firmy: Genoa Systems (Super VGA HiRes), Sigma Designs (Sigma VGA), ATI Technologies (VIP VGA), Video 7 (VEGA VGA), Everex Systems (EVGA) i Paradise (PVGA). Ceny tych kart kształtują się w granicach 400—600 $ (firmy tajwańskie oferują karty VGA już za 230 $).

Silna konkurencja na rynku powoduje, że poszczególni producenci wyposażają swoje sterowniki w liczne dodatkowe tryby graficzne (aż do rozdzielczości 1024 × × 768) oraz, oprócz CGA i EGA, emulację trybu pracy karty Hercules, a także drivery do najpopularniejszych programów (AutoCAD, GEM, Windows, Lotus 1-2-3, Symphony, Ventura Publisher i innych). Okazuje się, że nie wszystkie karty reklamowane jako zgodne z VGA w pełni odpowiadają temu standardowi. Obejmuje on

Karty graficzne produkcji firmy Everex



w sumie 17 trybów graficznych i tekstowych (w tym 5 nowych), wywoływanych jako procedury BIOS-a. Warto też zwrócić uwagę na inny aspekt zgodności: czy jest to zgodność programowa (na poziomie BIOS-a), czy sprzętowa (na poziomie rejestrów). Bliższa idealnej zgodności (z punktu widzenia oprogramowania) jest oczywiście zgodność sprzętowa, dodatkowym jej atutem jest większa szybkość generowania obrazu. Zgodność sprzętową można uzyskać np. stosując w karcie graficznej specjalizowany układ scalony TVGA firmy Trident Systems (znana firma Chip & Technologies produkuje jednoukładowe sterowniki VGA zapewniające zgodność tylko na poziomie BIOS-a). Podobny problem występował również w odniesieniu do ulepszonych sterowników EGA.

Pojawienie się standardu VGA oznacza zmierzch krótkotrwałego standardu EGA. Dodatkowy gwóźdź do trumny wbiła firma IBM zmieniając sposób transmisji sygnału ze sterownika do monitora. EGA i CGA współpracują z monitorami o wejściu cyfrowym RGB TTL (9-stykowym), VGA wymaga monitora z wejściem analogowym RGB (15-stykowym) ze względu na zwiększone pasmo częstotliwościowe sygnałów wizyjnych (rzędu 34 MHz). Kabel TTL współpracujący z kartą VGA musiałby mieć średnicę kciuka! Analogowy sposób przesyłu sygnału pozwala teoretycznie na uzyskanie na monitorze nieskończonej gamy kolorów, ograniczonej tylko rozdzielczością sterownika i rozmiarem jego pamięci obrazu. Nabywcy kosztownych monitorów EGA, którzy pragną zmienić standard na VGA, muszą więc kupić nowy monitor — chyba, że byli na tyle przewidujący, że zaopatrzyli się w tzw. monitor multisynchroniczny, pracujący w szerokim pasmie częstotliwości synchronizacji poziomej (minimum 32 kHz) i pionowej, posiadający zarówno wejścia analogowe, jak i cyfrowe. Monitory takie po raz pierwszy wprowadzono w roku 1985 (NEC Multisync, Sony Multiscan CPD 1302) i bardzo szybko zdobyły one dużą popularność, szczególnie w USA. Użycie monitora multisynchronicznego rozwiązuje problem zgodności zarówno ze wszystkimi obecnie istniejącymi standardami, jak i standardami o jeszcze większej rozdzielczości — np. 800 × 600, które ewentualnie mogą upowszechnić się w najbliższej przyszłości.

Najbardziej znanym producentem monitorów multisynchronicznych jest prawdopodobnie firma NEC (Multisync II, Multisync XL), z którą konkurują m.in. Princeton (Ultrasync), Sony (Multiscan CPD 1303), Mitsubishi (Diamond Scan), Magnavox (Multimode), Amdek (Amdek 1280). Spośród firm tajwańskich głównym potentatem jest firma TVM, znana również w Polsce (monitor multisynchroniczny MD-11 o rozdzielczości 800 × 600 można sprowadzić z Tajwanu za ok. 550 $ wliczając koszt transportu). W USA ceny monitorów multisynchronicznych kształtują się w granicach 550—3000 $.

Jakie są dalsze perspektywy rozwoju sterowników graficznych do mikrokomputerów osobistych? Odpowiedź wydaje się oczywista. Jak w każdej dziedzinie i w tej dominuje dążenie do wytwarzania produktów o coraz lepszych parametrach. W najbliższych latach można spodziewać się powszechniejszego wykorzystania procesorów i koprocesorów graficznych, jak np. Intel 82786 czy Texas Instruments TMS 34010, oraz stałego zwiększania obszaru pamięci obrazu zawartej na karcie sterownika.

Zdaniem autora przyszłość należy do kart mających wbudowane możliwości emulacji wielu trybów pracy. Dlatego też kupując nową kartę do komputera warto się zastanowić, czy dodatkowy wydatek poniesiony na lepszą (bardziej uniwersalną) kartę nie zaprocentuje kiedyś w sposób bardzo znaczący.

Od redakcji:
Artykuł pisany był we wrześniu 1988.

KOMPUTER W SZKOLE

10001 — ile to jest trójkowo?

Czasem zdarza się, że stajemy przed koniecznością przeliczenia liczby z jednego systemu na inny, np. liczbę w kodzie dziesiętnym trzeba zamienić na liczbę binarną, szesnastkową na dziesiętną itp. Jeżeli liczb tych jest niewiele, można to w miarę szybko policzyć na kawałku papieru. Gdy jest ich więcej czas potrzebny na niezbędne obliczenia wydaje się zbyt długi. W takich przypadkach pomocą służy niniejszy program przeznaczony na komputer ZX Spectrum (i zgodne z nim).

Program służy do zamiany liczby o dowolnej podstawie nie większej od 16 na liczbę o innej podstawie spełniającej taki sam warunek.

Obsługa programu jest bardzo prosta. Po uruchomieniu instrukcją RUN program zgłasza się pytaniem o podstawę liczby przeliczanej, a po jej uzyskaniu — o samą liczbę.

Podczas wprowadzania liczby akceptowane są tylko te klawisze, które w danej sytuacji mogą być użyte (np. gdy zadeklarujemy podstawę równą osiem przyjmowane będą tylko cyfry z zakresu 0—7, a gdy podstawą będzie liczba dwa, używać można tylko cyfr 0 i 1). Jeżeli podana wcześniej podstawa jest większa od dziesięciu zakres cyfr zwiększa się. W szczególności, gdy podstawa wprowadzanej liczby jest równa szesnaście, kolejnymi cyframi są 'A', 'B', 'C', 'D', 'E' i 'F'.

Trzecią i ostatnią daną wejściową jest wartość nowej podstawy, która także musi być nie większa niż 16.

W razie pomyłki błędnie wprowadzoną daną można skasować wciskając klawisz DELETE (CS&0). Zakończenie wprowadzania danych uzyskujemy używając klawisza ENTER.

Po uzyskaniu danych wejściowych i dokonaniu przeliczeń program wyświetla na ekranie obie liczby wraz z informacją, w jakim systemie liczbowym są one wyrażone.

Pewnym ograniczeniem programu jest to, że największa liczba, którą można za jego pomocą zamienić ma wartość 99999999 (dziesiętnie). Spowodowane jest to sposobem zapamiętywania liczb przez ZX Spectrum. Liczby większe zapamiętywane są przez interpreter BASIC-a w postaci zmiennoprzecinkowej, co może spowodować pewne przekłamania przy przetwarzaniu ciągów znaków. Nie jest to jednak ograniczenie szczególnie istotne, gdyż rzadko kiedy występuje potrzeba zamiany tak dużej liczby. Program zabezpieczony jest przed wprowadzeniem danej spoza zakresu, co sygnalizuje odpowiednim komunikatem.

Drugim ograniczeniem jest maksymalna wartość podstawy (nie większa niż 16). Użytkownik może to jednak zmienić we własnym zakresie — konieczne jest wprowadzenie nowych 'cyfr': 'G', 'H', 'I'... itd., w zależności od potrzeby. Oczywiście następstwem tego będzie konieczność zdefiniowania nowych znaków graficznych.

O zaletach programu każdy użytkownik może przekonać się samodzielnie, ale na uwagę zasługuje chyba prostota obsługi. Jak już wspomniałem program przeznaczony jest na komputery ZX Spectrum i zgodne z nim, jednak nie powinno sprawiać większych problemów przetłumaczenie go na inne dialekty BASIC-a. Należy zrezygnować ze znaków graficznych, a więc polskich liter i liczb oznaczających podstawę, lub zastąpić je w inny sposób. Ponadto trzeba zmienić sposób wczytywania danych z uwagi na nietypową instrukcję INKEY$. Ambitnym życzę powodzenia!

A teraz kilka słów o samym programie, jego strukturze i działaniu.

Instrukcja POKE w linii 40 powoduje przejście edytora w tryb 'C', czyli przyjmowania jedynie dużych liter. Linie 50—140 służą do wczytania kształtów dodatkowych znaków w obszar UDG oraz zdefiniowanie nowych cyfr.

Dalej następuje część programu realizująca pobieranie danych wejściowych. Zmienne 'podst1' i 'podst2' zawierają odpowiednio „starą" i „nową" podstawę liczby, natomiast w zmiennej 'liczba' umieszczana jest dziesiętna wartość liczby podanej przez użytkownika. Konieczność zamiany podawanych liczb na kod dziesiętny (dokonywanej przez linie 390—420) spowodowana jest tym, że interpreter BASIC-a operuje na liczbach o tej podstawie. Zamiana liczby z kodu dziesiętnego na liczbę o podstawie 'podst2' dokonuje się w liniach 540—650. Pod zmienną c$ podstawiane są kolejne cyfry wynikowej liczby, które razem tworzą liczbę zawartą w zmiennej m$. Linie 670—680 umożliwiają wyświetlenie na ekranie końcowego wyniku działania programu.

Wszystkim tym, których zainteresował powyższy program i zdecydowali się go przepisać życzę przyjemnej zabawy i nauki.

Adam Drabik

```
 10 REM ************************
 20 REM PODSTAWY (c) ADAM DRABIK
 25 REM      (ZX Spectrum)
 30 REM ************************
 35 REM
 36 REM  UWAGA! litery podkreślone oznaczają znaki w trybie
    graficznym, znak u oznacza spację
 39 REM
 40 POKE 23658,8
 45 PAPER 1: INK 7: BORDER 1: CLS
 50 FOR n=USR "a" TO USR "a"+135
 60 READ a
 70 POKE n,a
 80 NEXT n
 90 LET a=10
100 LET b=11
110 LET c=12
120 LET d=13
130 LET e=14
140 LET f=15
170 PLOT 0,38: DRAW 255,0
180 PRINT AT 18,0;"DELuu-kasowanieuostatniegouznaku",AT 20,0;
    "ENTER-koniecuwprowadzaniaudanych"
190 LET p$=""
195 PRINT AT 3,0;"Podajupodstawęu(2-16)"
200 PRINT AT 5,0;p$; FLASH 1;"u"; FLASH 0;" "
205 BEEP .1,20
210 LET a$=INKEY$
220 IF a$="" THEN GO TO 210
230 IF CODE a$=12 AND p$<>"" THEN GO SUB 750: GO TO 200
240 IF CODE a$=13 AND p$<>"" THEN BEEP .3,10: GO TO 270
250 IF (a$>="0" AND a$<="9") AND LEN p$<3 THEN LET p$=p$+a$:
    GO TO 200
260 GO TO 210
270 LET podst1=VAL p$
280 IF podst1>1 AND podst1<17 THEN PRINT AT 5,0;p$,"u": GO TO 300
```

```
290 GO SUB 710: GO TO 210
300 LET p$=""
310 PRINT AT 7,0;"Podaj␣liczbQ?"
320 PRINT AT 9,0;p$; FLASH 1;"␣"; FLASH 0;"␣"
325 BEEP .1,20
330 LET a$=INKEY$
340 IF a$="" THEN GO TO 330
350 IF CODE a$=12 AND p$<>"" THEN GO SUB 750: GO TO 320
360 IF CODE a$=13 AND p$<>"" THEN BEEP .3,10: GO TO 383
370 IF ((a$>="0" AND a$<="9") OR (a$>="A" AND a$<="F")) AND LEN
    p$<31 THEN IF VAL a$<podst1 THEN LET p$=p$+a$: GO TO 320
380 GO TO 330
383 PRINT AT 9,0;p$;"␣"
385 LET l$=p$
390 LET liczba=0
400 FOR n=0 TO LEN p$-1
410 LET liczba=liczba+VAL p$(n+1)*podst1↑(LEN p$-1-n)
420 NEXT n
425 IF liczba>99999999 THEN GO SUB 780: GO TO 320
430 LET p$=""
435 PRINT AT 11,0;"Podaj␣nowP␣podstawQ? (2-16)"
440 PRINT AT 13,0;p$; FLASH 1;"␣"; FLASH 0;" "
445 BEEP .1,20
450 LET a$=INKEY$
460 IF a$="" THEN GO TO 450
470 IF CODE a$=12 AND p$<>"" THEN GO SUB 750: GO TO 440
480 IF CODE a$=13 AND p$<>"" THEN BEEP .3,10: GO TO 510
490 IF (a$>="0" AND a$<="9") AND LEN p$<31 THEN LET p$=p$+a$:
    GO TO 440
500 GO TO 450
510 LET podst2=VAL p$
520 IF podst2>1 AND podst2<17 THEN PRINT AT 13,0;p$;"␣": GO TO 540
530 GO SUB 710: GO TO 450
540 LET m$=""
545 LET koniec=0
550 IF liczba<podst2 THEN LET c$=STR$ liczba: LET koniec=1: GO TO 570
560 LET c$=STR$ INT (liczba-INT (liczba/podst2)*podst2)
565 LET liczba=INT (liczba/podst2)
570 IF c$="10" THEN LET c$="A"
580 IF c$="11" THEN LET c$="B"
590 IF c$="12" THEN LET c$="C"
600 IF c$="13" THEN LET c$="D"
610 IF c$="14" THEN LET c$="E"
620 IF c$="15" THEN LET c$="F"
630 LET m$=c$+m$
650 IF koniec=0 THEN GO TO 550
660 CLS
670 PRINT AT 9,0;l$;CHR$ (142+podst1);"=";AT 11,0; "="; m$;
    CHR$ (142+podst2)
680 PRINT AT 21,0;"NaciSnij␣jakiS␣klawisz..."
690 PAUSE 0
695 CLS
700 GO TO 170
710 PRINT AT 16,21; FLASH 1; PAPER 2; BRIGHT 1;"BRQdne␣dane"
720 BEEP .5,0: BEEP .25,15: BEEP .5,0: BEEP .25,15: BEEP .5,0:
    BEEP .25,15
730 PRINT AT 16,21;"␣␣␣␣␣␣␣␣␣␣␣"
740 RETURN
750 IF LEN p$>1 THEN LET p$=p$( TO LEN p$-1): RETURN
760 LET p$=""
770 RETURN
780 PRINT AT 16,12; FLASH 1; PAPER 2; BRIGHT
1;"Liczba␣poza␣zakresem"
790 BEEP .5,0: BEEP .25,15: BEEP .5,0: BEEP .25,15: BEEP .5,0:
    BEEP .25,15
800 PRINT AT 16,12;"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"
810 RETURN
840 DATA 0,0,0,56,8,56,32,56
850 DATA 0,0,0,56,8,24,8,56
860 DATA 0,0,0,32,40,56,8,8
870 DATA 0,0,0,56,32,56,8,56
880 DATA 0,0,0,56,32,56,40,56
890 DATA 0,0,0,56,8,8,8,8
900 DATA 0,0,0,56,40,56,40,56
910 DATA 0,0,0,56,40,56,8,56
920 DATA 0,0,0,46,42,42,42,46
930 DATA 0,0,0,36,36,36,36,36
940 DATA 0,0,0,46,34,46,40,46
950 DATA 0,0,0,46,34,38,34,46
960 DATA 0,0,0,40,42,46,34,34
970 DATA 0,0,0,46,40,46,34,46
980 DATA 0,0,0,46,40,46,42,46
990 DATA 0,0,56,4,60,68,60,8
1000 DATA 0,0,56,68,120,64,60,16
1010 DATA 0,16,24,16,48,16,12,0
1020 DATA 8,16,56,64,56,4,120,0
```


ASEMBLER GENS3

Część VIII

Przedstawiony poniżej przykład programu ma na celu praktyczne zapoznanie czytelników z niektórymi typowymi problemami programowania w języku asemblera. Nie było zamierzeniem autora opisywanie jakiegoś wyszukanego programu realizującego konkretne zadanie. Zadaniem nadrzędnym było stworzenie pewnego „szablonu", który czytelnicy będą mogli modyfikować i rozwijać w zależności od własnych potrzeb. Co prawda robocza nazwa „Mini-monitor" określa z grubsza jego funkcje, ale łatwo mogą one zostać zmienione. Dla zrozumienia procesu tworzenia programu, ze względów dydaktycznych podzielono wydruk na kilkanaście dość niezależnych funkcjonalnie części. Będą one kolejno szczegółowo omawiane. Jeśli po dokładnym przeanalizowaniu tekstu czytelnik będzie czuł się na siłach napisać własny program, autor będzie przekonany, że cel został osiągnięty.

Założenia

W celu jasnego sprecyzowania problemu na wstępie przyjęto następujące założenia:

1. Program powinien w minimalnej wersji umożliwiać zrealizowanie następujących elementarnych funkcji:
   a) przenoszenie bloków danych w pamięci operacyjnej,
   b) zapełnianie zadanego obszaru pamięci stałą wartością,
   c) dodawanie i odejmowanie liczb całkowitych w zakresie (0—65535),
   d) zamianę liczb w postaci dziesiętnej, binarnej i szesnastkowej w każdej kombinacji,
   e) zapis i odczyt dowolnego bloku danych na taśmie magnetofonowej,
   f) wyświetlenie zawartości pamięci w postaci dziesiętnej, szesnastkowej i znaków ASCII z podaniem adresów w postaci dziesiętnej i szesnastkowej,
   g) zmianę zawartości pamięci operacyjnej,
   h) uruchomienie programu maszynowego,
   i) wyświetlenie zawartości wszystkich rejestrów.
2. Wszystkie zlecenia będą identyfikowane przez podanie jednej litery. W każdym zleceniu może wystąpić maksymalnie do trzech parametrów. Parametry będą oddzielane znakiem przecinka, a zlecenie kończy naciśnięcie klawisza ⟨ENTER⟩. Naciśnięcie klawisza ⟨DELETE⟩ powoduje kasowanie aktualnie wprowadzanego zlecenia. Powrót z programu następuje po naciśnięciu klawisza ⟨EDIT⟩.
3. Każda z liczb będąca parametrem może być wprowadzona w dowolnej postaci, jeśli zostanie poprzedzona odpowiednim do typu znakiem:
   & dla liczb dziesiętnych,
   # dla liczb szesnastkowych,
   % dla liczb binarnych,
   $ dla znaków ASCII.

4. Program zachowuje wszystkie rejestry, używa własnego stosu i bufora do wprowadzania danych.
5. Program będzie wykorzystywał niektóre standardowe procedury umieszczone w ROM, a także kilka zmiennych systemowych.

Wykorzystane zmienne i procedury zewnętrzne

Przedstawiony poniżej fragment wydruku zawiera deklaracje użytych zmiennych systemowych oraz procedur zawartych w ROM. Poniżej znajdują się rezerwacje zmiennych użytych wewnątrz programu.

KOMP — zmienna wykorzystywana do asemblacji warunkowej, ponieważ publikowane w tym odcinku fragmenty nie stanowią jeszcze kompletnego programu (patrz wydruk nr 5 i 6).

LASTK — program obsługi przerwania maskowanego nadaje tej zmiennej wartość kodu aktualnie wciśniętego klawisza.

FLAGS2 — bit 3 = „0" oznacza pracę w trybie „L" (małe litery), bit 3 = „1" oznacza pracę w trybie „C" (duże litery).

PRINTA — procedura wyprowadzająca zawartość akumulatora na aktualne urządzenie wyjścia.

CLS — procedura kasuje zawartość ekranu.

AT — procedura ustawia pozycję wydruku. Numer wiersza w rejestrze B, numer kolumny w rejestrze C. (Lewy górny róg ekranu to pozycja (24,33)).

SCROLL — procedura przesuwa zawartość ekranu o jeden wiersz w górę.

ERASE — procedura kasuje zawartość ekranu od danego wiersza włącznie w dół ekranu. Numer wiersza w rejestrze B (najniższy wiersz ma numer 1).

OPEN — procedura otwiera aktualny kanał wyjściowy. Numer kanału należy podać do akumulatora (ekran ma numer 2, drukarka 3).

STKBC — umieszcza na stosie kalkulatora (w postaci zmiennoprzecinkowej) zawartość par rejestrów BC.

PRTFP — wyprowadza na ekran ostatnio umieszczoną na stosie kalkulatora liczbę.

Dalej od linii 200 do linii 380 następują rezerwacje pamięci dla używanych w programie zmiennych. Na przechowanie zawartości wszystkich rejestrów (główny zestaw rejestrów ze wskaźnikiem jeden, a alternatywny ze wskaźnikiem dwa). Dalej jedna komórka WSKAZN, na pomocnicze wskaźniki. Zmienna ADRES, w której będzie przechowywany adres aktualnie wykonywanego zlecenia. Trzy dwubajtowe zmienne PARAM1, PARAM2, PARAM3 na przechowywanie parametrów zlecenia. Na końcu rezerwacja pamięci na bufor danych i stos.

```
 10 ;MINI-MONITOR CZ.1
 20 ;Kurs asemblera Gens3 cz.4
 30 ;(C) Tadeusz Basista 1988
 40 ;-------------------------
 50 KOMPL   EQU  0
 60         ORG  60000
 70         JP   START
 80
```

```
 90 LASTK  EQU  23560
100 FLAGS2 EQU  23658
110 PRINTA EQU  #10
120 CLS    EQU  #0D6B
130 AT     EQU  #0DD9
140 SCROLL EQU  #0DFE
150 ERASE  EQU  #0E44
160 OPEN   EQU  #1601
170 STKBC  EQU  #2D2B
180 PRTFP  EQU  #2DE3
190 ;-----------------------------
200 REJSP  DEFS 2
210 REJIR  DEFS 2
220 REJIY  DEFS 2
230 REJIX  DEFS 2
240 REJHL2 DEFS 2
250 REJDE2 DEFS 2
260 REJBC2 DEFS 2
270 REJAF2 DEFS 2
280 REJHL1 DEFS 2
290 REJDE1 DEFS 2
300 REJBC1 DEFS 2
310 REJAF1 DEFS 2
320 WSKAZN DEFS 1
330 ADRES  DEFS 2
340 PARAM1 DEFS 2
350 PARAM2 DEFS 2
360 PARAM3 DEFS 2
370 BUFOR  DEFS 100
380 STOS   DEFB #80
```

Przechowanie i odtworzenie zawartości rejestrów

Pierwszą czynnością realizowaną przez program powinno być przechowanie zawartości rejestrów, a ostatnią jej odtworzenie. Narzucające się rozwiązanie bezpośredniego wpisywania zawartości rejestrów do zarezerwowanych komórek pamięci zastąpiono w poniższym przykładzie procedurą wykorzystującą mechanizm działania stosu. Na wstępie zostaje zapamiętany adres wskaźnika stosu, a następnie rejestr ten jest ustawiany na adresie pierwszej wolnej komórki po obszarze przeznaczonym na przechowywanie zawartości rejestrów (zmienna WSKAZN). Teraz umieszczanie na stosie zawartości kolejnych rejestrów będzie równoznaczne z wpisywaniem ich w kolejne zmienne. Przypomnieć należy tu, że zgodnie z organizacją stosu następuje to w kierunku niższych adresów, tzn. za każdym razem aktualna wartość wskaźnika stosu jest zmniejszana o dwa. Po zakończeniu tych czynności wskaźnik stosu jest ustawiany na ostatnim bajcie zarezerwowanego przez nas obszaru i jest wywoływany program główny. Zakończenie programu głównego powoduje powrót do linii 660, gdzie realizowane są kolejno dokładnie odwrotne czynności. Wskaźnik stosu jest ustawiany na pierwszym adresie obszaru przeznaczonego na przechowywanie zawartości rejestrów. Teraz „zdejmowanie" ze stosu do odpowiednich par rejestrów zawartości pamięci jest równoznaczne z wpisywaniem do nich uprzednio zapamiętanych wartości. Tym razem wskaźnik stosu po każdej z tych czynności zostaje zwiększony o dwa. W końcu zostaje odtworzona pierwotna zawartość wskaźnika stosu. Przedstawione rozwiązanie gwarantuje nam bezkolizyjny powrót do systemu, jest dość uniwersalne i oszczędne pod względem długości kodu (większość instrukcji jest jednobajtowa).

```
390 ;-----------------------------
400 ;zachowanie rejestrow
410 START  LD   (REJSP),SP
420        LD   SP,WSKAZN
430        PUSH AF
440        PUSH BC
450        PUSH DE
460        PUSH HL
470        EX   AF,AF'
480        EXX
490        PUSH AF
500        PUSH BC
```

```
510          PUSH DE
520          PUSH HL
530          EX   AF,AF'
540          EXX
550          PUSH IX
560          PUSH IY
570          LD   A,I
580          LD   H,A
590          LD   A,R
600          LD   L,A
610          PUSH HL
620          LD   SP,STOS
630          CALL PROGR1
640
650 ;odtworzenie rejestrow
660          LD   SP,REJIR
670          POP  HL
680          LD   A,L
690          LD   R,A
700          LD   A,H
710          LD   I,A
720          POP  IY
730          POP  IX
740          EX   AF,AF'
750          EXX
760          POP  HL
770          POP  DE
780          POP  BC
790          POP  AF
800          EX   AF,AF'
810          EXX
820          POP  HL
830          POP  DE
840          POP  BC
850          POP  AF
860          LD   SP,(REJSP)
870          RET
880    ----------------------------
```

Obsługa klawiatury

Procedura obsługi klawiatury wykorzystywana w głównej pętli programowej jest poprzedzona krótką procedurą testującą zawartość akumulatora. Warto ją opisać dokładnie, ponieważ jest to jeden z najkrótszych sposobów ustalenia czy zawartość akumulatora jest liczbą znajdującą się w zadanym przedziale. Na znaki określające typ parametru wybrano #, $, %, &. Mają one kolejne wartości kodów 35, 36, 37 i 38. Najpierw porównujemy czy zawartość akumulatora jest równa pierwszej liczbie należącej do tego przedziału. Jeżeli jest mniejsza następuje opuszczenie procedury z wartością wskaźnika CY = 1. Następnie porównujemy zawartość akumulatora z pierwszą liczbą następującą po badanym przedziale (w tym przypadku kod apostrofu o wartości 39). Jeśli teraz zawartość akumulatora jest większa lub równa wartości tego kodu wskaźnik CY = 0. Przed wyjściem z procedury wskaźnik CY jest zmieniany na przeciwny. Widać stąd, że jeśli zawartość akumulatora leży poza badanym przedziałem to CY = 1, a jeśli wewnątrz to CY = 0.

W procedurze odczytu klawiatury wykorzystano adresowaną przy pomocy rejestru IY zmienną systemową FLAGS (adres 23611). Ustawienie trzeciego bitu powoduje, że klawisz zostanie odczytany jako znak. Następnie testowana jest wartość bitu piątego. Jeśli jego wartość jest równa jeden znaczy to, że został naciśnięty nowy klawisz. Jego kod jest pobierany ze zmiennej systemowej LASTK do akumulatora. Teraz sprawdzamy, czy jest to kod klawisza ⟨DELETE⟩. Jeśli tak, to wychodzimy z procedury (zauważmy dodatkowo, że w tym przypadku CY = 1). Następnie (linia 1080) sprawdzamy, czy jest to kod klawisza zmiany typu parametru. Jeśli nie, to wychodzimy z procedury z wartością CY = 0. Jeśli tak, to wskaźnik typu parametru jest zapamiętany w nie wykorzystywanej zmiennej systemowej o adresie 23681. Następnie pozycja kursora jest cofana o jedną pozycję i wyprowadzany znak typu parametru, z tym że w przypadku argumentów typu dziesiętnego jest on zastępowany spacją. Przed wyjściem z procedury kasujemy wskaźnik CY, tak więc CY = 1 jedynie w przypadku klawisza ⟨DELETE⟩.

```
 890 ;test czy klawisz zmiany
 900 ;trybu.Jesli tak to CY=0
 910 TESTTR CP   "#"
 920        RET  C
 930        CP   "'"
 940        CCF
 950        RET
 960 ;----------------------------
 970 ;odczyt klawisza
 980 ;jesli A=delete to CY=1
 990 KLAW1  SET  3,(IY+1)
1000 KLAW2  BIT  5,(IY+1)
1010        JR   Z,KLAW2
1020        RES  5,(IY+1)
1030        LD   A,(LASTK)
1040        CP   #0C         ;del
1050        CCF
1060        RET  Z
1070
1080        CALL TESTTR
1090        CCF
1100        RET  NC
1110
1120        LD   (IY+71),A
1130        PUSH AF
1140        LD   A,8
1150        RST  PRINTA
1160        POP  AF
1170 KLAW3  PUSH AF
1180        LD   A,(IY+71)
1190        CP   "&"
1200        JR   C,KLAW4
1210
1220        LD   A," "
1230 KLAW4  RST  PRINTA
1240        POP  AF
1250        AND  A
1260        RET
1270   ----------------------------
```

Wyprowadzanie danych

Pierwsza procedura DHCOD zamienia zawartość akumulatora z przedziału (0—15) na kod odpowiedniego znaku ASCII, tzn. na kody „0"—„9" oraz „A"—„F". Zastosowano tu podobny chwyt jak w procedurze TESTTR. Procedura PR1HEX wyprowadza szesnastkowo zawartość akumulatora. W tym celu najpierw „obcina" się cztery młodsze bity, następnie przesuwa w prawo zawartość akumulatora i po dokonaniu konwersji wyprowadza. Następnie obcinane są cztery starsze bity i po konwersji bezpośrednio wyprowadzana zawartość akumulatora.

Procedura PR2HEX korzystająca z poprzedniej procedury wyprowadza szesnastkowo zawartość pary rejestrów BC, dodatkowo po niej jest wyprowadzana jedna spacja.

Procedura PRDHBC wyprowadza zawartość pary rejestrów BC najpierw szesnastkowo poprzedzoną jedną spacją i znakiem #, następnie dziesiętnie poprzedzoną jedną spacją.

Ponieważ w przedstawianym programie będziemy wielokrotnie przy redagowaniu wydruku ustawiali jego pozycję, dla zaoszczędzenia długości programu będziemy

stosowali procedurę PRTAT. Ponieważ wszystkie wydruki będą się ukazywały w trzecim od dołu wierszem ekranu, będzie ustawiana jedynie pozycja kolumny. Parametr ten będzie umieszczony jako bajt zdefiniowany bezpośrednio za wywołaniem procedury PRTAT (patrz wydruk nr 6 i nr 7). W celu pobrania tego parametru wystarczy zamienić zawartość wskaźnika stosu z parą rejestrów HL, ponieważ po wywołaniu procedury na stosie umieszczany jest adres powrotu, w tym przypadku identyczny z adresem definiowanego bajtu. W celu zagwarantowania poprawnego powrotu do programu przed wyjściem z procedury należy zwiększyć ten adres o jeden. Taki sposób przekazywania parametrów jest często stosowany na poziomie asemblera, choć jak widać powoduje on przemieszanie struktury danych i instrukcji programowych. Nie jest to przeszkoda, jeśli jesteśmy w posiadaniu wersji źródłowej programu, natomiast może dość skutecznie zniechęcić do deasemblacji kodu wynikowego.

```
1280 ;zamiana na Postac szesn.
1290 DHCOD  ADD  A,48
1300        CP   58
1310        RET  C
1320        ADD  A,7
1330        RET
1340 ;-------------------------
1350 ;wyProw.szesn.akumul.
1360 PR1HEX PUSH AF
1370        AND  %11110000
1380        RRCA
1390        RRCA
1400        RRCA
1410        RRCA
1420        CALL DHCOD
1430        RST  PRINTA
1440        POP  AF
1450        AND  %00001111
1460        CALL DHCOD
1470        RST  PRINTA
1480        RET
1490 ;-------------------------
1500 ;wyProw.szesn.rej.BC
1510 PR2HEX PUSH BC
1520        LD   A,B
1530        CALL PR1HEX
1540        POP  BC
1550        LD   A,C
1560        CALL PR1HEX
1570        LD   A," "
1580        RST  PRINTA
1590        RET
1600 ;-------------------------
1610 ;wyProw.dzies.rej.BC
1620 PR2DEC CALL STKBC
1630        CALL PRTFP
1640        RET
1650 ;-------------------------
1660 ;wyProw.dzies.i szesn.BC
1670 PRDHBC PUSH BC
1680        LD   A," "
1690        RST  PRINTA
1700        LD   A,"#"
1710        RST  PRINTA
1720        POP  BC
1730        PUSH BC
1740        CALL PR2HEX
1750        LD   A," "
1760        RST  PRINTA
1770        POP  BC
1780        CALL PR2DEC
1790        RET
1800 ;-------------------------
1810 ;ustaw.Poz.druku.Kol.w C
1820 PRTAT  EXX
1830        EX   (SP),HL
1840        LD   C,(HL)
1850        LD   B,3
1860        INC  HL
1870        EX   (SP),HL
1880        CALL AT
1890        EXX
1900        PET
1910  -------------------------
```

Menu programu

Wyboru odpowiedniego zlecenia, tzn. identyfikacji nazwy zlecenia, ustalenia liczby niezbędnych parametrów i adresu odpowiedniego podprogramu dokonuje procedura WYBOR1. Wykorzystuje ona tablicę zawierającą po trzy kolejne, na każde zlecenie, deklaracje: kodu klawisza, liczby wymaganych parametrów i adresu podprogramu realizującego zlecenie. Tablica jest zakończona nie występującym w jej wnętrzu bajtem równym zero. Działanie procedury jest następujące. Do pary rejestrów HL jest wpisywany adres początku tablicy. Kod naciśniętego klawisza zapamiętywany jest w rejestrze C. Następnie sprawdza się czy wykryto koniec tablicy (linia 2460). Jeśli tak, to następuje wyjście z procedury. Zauważmy dodatkowo, że w tym przypadku wskaźnik CY = 0. Jeśli nie natrafiono na koniec tablicy kod zlecenia w rejestrze C jest porównywany z pobranym do akumulatora kodem z tablicy. Jeśli nie są identyczne adres w HL jest zwiększany o cztery i przeszukiwanie kontynuowane. Jeśli kody są identyczne następuje pobranie danych z tablicy. W rejestrze D znajdzie się kod zlecenia, w rejestrze E liczba parametrów, w parze rejestrów BC adres podprogramu. Przed wyjściem z procedury w tym przypadku wskaźnik CY jest ustawiany.

Użyta w linii 1930 pseudoinstrukcja asemblacji warunkowej spowoduje, że w aktualnej wersji zostaną asemblowane jedynie dane dotyczące dwóch ostatnich zleceń.

```
1920 TABINS
1930        IF   KOMPL
1940
1950        DEFM "B"
1960        DEFB 3
1970        DEFW BLOKI
1980
1990        DEFM "C"
2000        DEFB 3
2010        DEFW STALA
2020
2030        DEFM "+"
2040        DEFB 2
2050        DEFW DODAJ
2060
2070        DEFM "-"
2080        DEFB 2
2090        DEFW ODEJM
2100
2110        DEFM "E"
2120        DEFB 1
2130        DEFW ZAMIEN
2140
2150        DEFM "L"
2160        DEFB 1
2170        DEFW ODCZYT
2180
2190        DEFM "S"
2200        DEFB 3
2210        DEFW ZAPIS
2220
2230        DEFM "M"
2240        DEFB 1
2250        DEFW PAMIEC
2260
2270        DEFM "J"
2280        DEFB 1
2290        DEFW SKOK
2300
2310        END
2320
2330        DEFM "R"
2340        DEFB 0
2350        DEFW REJEST
2360
2370        DEFB 7         ;edit
2380        DEFB 0
2390        DEFW POWROT
2400
```

```
2410          DEFB 0
2420 ;--------------------------
2430 WYBOR1 LD   HL,TABINS
2440        LD   C,A
2450 WYBOR2 LD   A,(HL)
2460        AND  A
2470        RET  Z
2480        CP   C
2490        JR   Z,WYBOR3
2500        INC  HL
2510        INC  HL
2520        INC  HL
2530        INC  HL
2540        JR   WYBOR2
2550
2560 WYBOR3 LD   D,(HL)
2570        INC  HL
2580        LD   E,(HL)
2590        INC  HL
2600        LD   C,(HL)
2610        INC  HL
2620        LD   B,(HL)
2630        LD   (ADRES),BC
2640        SCF
2650        RET
2660 --------------------------
```

Główna pętla programowa

Po pierwszym uruchomieniu programu wskaźnik typu parametrów jest ustawiany wstępnie na dziesiętne. Następnie zostaje jako urządzenie wyjściowe przydzielony ekran. Główna pętla programowa rozpoczyna się od etykiety PROGR2. Tu ustawiany jest tryb dużych liter, zawartość ekranu przesuwana o jedną linię do góry i program zgłasza się znakiem „>" oczekując na wprowadzenie zlecenia. Komentarza wymaga sposób powrotu do początku tej pętli. Adres etykiety PROGR2 jest zapamiętywany na stosie, tak że po testowaniu jakichkolwiek warunków wystarczy wykorzystanie instrukcji powrotu warunkowego, zamiast instrukcji skoku, jeśli chcemy wrócić do tego miejsca. Jest to oczywiście sprawa stylu programowania, ale autor jest zdania, że jest to bardziej eleganckie rozwiązanie (a na pewno krótsze) niż nadużywanie instrukcji skoku.

Następnie od etykiety PROGR3 oczekiwane jest wprowadzenie zlecenia. Naciśnięcie klawisza ⟨DELETE⟩ spowoduje powrót do głównej pętli, naciśnięcie klawisza nie oznaczającego żadnego zlecenia powoduje dalsze oczekiwanie. Decyduje o tym odpowiednio ustawiony przez procedury KLAW1 i WYBOR1 wskaźnik CY. Jeśli podano poprawny kod zlecenia jest on wyprowadzany, a po nim znak dwukropka i znak aktualnie ustawionego typu parametrów. Następnie jest sprawdzane czy zlecenie potrzebuje parametrów. Jeśli nie, następuje skok do etykiety PROGR4 i wykonanie odpowiedniego podprogramu. Jeśli parametry są wymagane dokonuje tego procedura USTPAR (w tym przypadku linie te nie będą asemblowane). Pobrane parametry są dodatkowo zapamiętywane w dwóch kompletach rejestrów: raz w HL', DE', BC', drugi raz w IX, BC, DE.

```
2670
2680 PROGR1
2690        LD   (IY+71),"&"
2700        LD   A,2
2710        CALL OPEN
2720        CALL CLS
2730 PROGR2
2740        LD   A,8
2750        LD   (FLAGS2),A
2760        LD   HL,PROGR2
2770        PUSH HL
2780        CALL SCROLL
2790        CALL PRTAT
2800        DEFB 33
2810        LD   A,">"
```

```
2820        RST  PRINTA
2830 PROGR3
2840        CALL KLAW1
2850        RET  C
2860
2870        CALL WYBOR1
2880        JR   NC,PROGR3
2890
2900        PUSH DE
2910        LD   A,D
2920        RST  PRINTA
2930        LD   A,":"
2940        RST  PRINTA
2950        CALL KLAW3
2960        POP  DE
2970        LD   A,E
2980        AND  A
2990        JR   Z,PROGR4
3000
3010        IF   KOMPL
3020
3030        LD   E,A
3040        CALL USTPAR
3050        RET  C
3060
3070        END
3080
3090        EXX
3100        LD   HL,(PARAM1)
3110        LD   DE,(PARAM2)
3120        LD   BC,(PARAM3)
3130        PUSH HL
3140        PUSH HL
3150        PUSH DE
3160        EXX
3170        POP  DE
3180        POP  BC
3190        POP  IX
3200 PROGR4 LD   HL,(ADRES)
3210        JP   (HL)
3220 ;--------------------------
3230 POWROT
3240        POP  HL
3250        RET
3260 ;--------------------------
3270 REJEST LD   HL,WSKAZN
3280        LD   D,3
3290 REJES1 PUSH DE
3300        LD   E,4
3310 REJES2 PUSH DE
3320        PUSH HL
3330        CALL SCROLL
3340        CALL PRTAT
3350        DEFB 31
3360        POP  HL
3370        DEC  HL
3380        LD   B,(HL)
3390        DEC  HL
3400        LD   C,(HL)
3410        PUSH HL
3420        CALL PRDHBC
3430        POP  HL
3440        POP  DE
3450        DEC  E
3460        JR   NZ,REJES2
3470        PUSH HL
3480        CALL SCROLL
3490        POP  HL
3500        POP  DE
3510        DEC  D
3520        JR   NZ,REJES1
3530        RET
3540 ;--------------------------
```

Realizacja podprogramów

W przedstawionym przykładzie zrealizowano dwa zlecenia: powrót do systemu (realizowany po naciśnięciu klawisza ⟨EDIT⟩) oraz wyświetlenie zawartości rejestrów. Pierwszy podprogram jest oczywisty. W drugim zorganizowane są dwie pętle: zewnętrzna od etykiety REJES1 i wewnętrzna od etykiety REJES2. Wynika to ze sposobu wyprowadzenia rejestrów.

AF, BC, DE, HL

AF', BC', DE', HL'

IX, IY, IR, SP

Każda grupa jest oddzielona pustą linią na ekranie.

TADEUSZ BASISTA

SILNIKI KROKOWE

Silniki krokowe są urządzeniami, które prawie zawsze można znaleźć w otoczeniu komputera. Każda, nawet najprostsza drukarka posiada ich zwykle kilka. A już obowiązkowo znajdują się w każdej stacji dysków montowanej w mikrokomputerach. Postęp techniczny w dziedzinie konstrukcji tych silników jest mało spektakularny, wręcz przeciwnie, stwarza on warunki do zapominania o ich istnieniu. Coraz silniejsze magnesy trwałe umożliwiają osiąganie tych samych momentów napędowych przy coraz mniejszych gabarytach, a nowoczesne samosmarne łożyska pozwalają praktycznie zapominać o ich konserwacji. Dobrze, gdy „zapominamy" o nich pamiętając na czym opiera się zasada ich pracy...

Już sama nazwa sugeruje, że silnik ten różni się od zwykłego silnika. Silnik krokowy jest silnikiem, w którym kąt obrotu wału nie może być dowolny. Zależnie od konstrukcji wał silnika w pojedynczym kroku może obrócić się o kąt nie przekraczający zwykle 120°. Aby wykonać pełny obrót, silnik musi wykonać kilka do kilkudziesięciu kroków. Jest to więc silnik o skwantowanym skoku, idealnie nadający się do sterowania za pomocą układów logicznych. Dla zapewnienia prawidłowej pracy wymaga on podania na swoje uzwojenia odpowiednio przesuniętych w fazie napięć zasilających, do których wytworzenia konieczne są układy sterujące wykonane w oparciu o układy logiczne. Podając na uzwojenia silnika odpowiednie sekwencje sterujące, możemy uzyskać ruch w jedną lub drugą stronę, przy czym kąt obrotu może być bardzo łatwo kontrolowany przez kontrolę ilości impulsów podawanych na uzwojenia silnika.

Na rysunkach przedstawiono uproszczony model silnika krokowego, tłumaczący jego działanie. Rysunek „a" przedstawia pozycję wyjściową, w której wirnik silnika utrzymywany jest poprzez pole magnetyczne, wytworzone w uzwojeniach stojana. Wirnik jest w istocie magnesem stałym, sztywno przymocowanym na wale silnika. Na rysunku przedstawiono przypadek najprostszy, gdy wirnik składa się z jednego magnesu stałego, posiadającego tylko jedną parę biegunów. W praktyce spotyka się silniki, w których wirnik posiada większą ilość par biegunów.

Liczba nabiegunników stojana najczęściej jest parzysta, jednak nie jest to reguła. Należy tutaj wyjaśnić, że nie stoi to absolutnie w sprzeczności z prawami fizyki, gdyż w przypadku stojana nie wszystkie nabiegunniki muszą być jednocześnie biegunami magnetycznymi, jedne mogą wytwarzać silniejsze zaś inne słabsze pole magnetyczne, zależnie od prądu przepływającego przez ich uzwojenia. Jednak jeżeli strumień magnetyczny zobrazować liczbą linii sił pola magnetycznego, to liczba linii wychodzących ze stojana jest równa liczbie linii, które do niego wnikają, gdyż inaczej mielibyśmy do czynienia z monopolem magnetycznym czyli wyodrębnionym pojedynczym biegunem.

Przełączniki P_1 i P_2 umożliwiają zmianę kierunku prądu w uzwojeniach stojana, a tym samym zmianę kierunku wytwarzanych przez te uzwojenia strumieni magnetycznych. Kierunek strumienia magnetycznego można określić korzystając z reguły prawej dłoni (przypomnę, że jeżeli wyprostowany kciuk prawej ręki będzie wskazywał umowny kierunek przepływu prądu, czyli od plusa do minusa, to zgięte palce prawej ręki wskażą kierunek powstających linii sił pola magnetycznego). Zmiana kierunku strumienia magnetycznego, w którymkolwiek uzwojeniu, spowoduje obrót wirnika. Na rysunku „b" pokazano sytuację po zmianie kierunku strumienia wytwarzanego przez uzwojenie A, co powoduje obrót wirnika przeciwny do ruchu wskazówek zegara o kąt 90°. Jeżeli teraz nastąpi przełączenie przełącznika P_2, to silnik obróci się o dalsze 90° w tę samą stronę. Jeżeli zamiast przełącznika P_2 ponownie przełączymy przełącznik P_1, to silnik powróci do położenia wyjściowego, co może być zapoczątkowaniem obrotu w drugą stronę. W układzie rzeczywistym, funkcję przełączników realizują klucze półprzewodnikowe sterowane z odpowiedniego układu logicznego, zapewniającego uzyskanie pożądanego kierunku obrotów silnika.

W praktyce układ logiczny musi poza tym zapewniać odpowiednio długi czas pomiędzy poszczególnymi „krokami", wykonanymi przez silnik. Czas ten jest potrzebny na wytłumienie drgań wirnika oraz na to, by prąd w cewkach sterujących mógł uzyskać nominalną wartość. Dynamiczne własności silników krokowych ograniczają maksymalną szybkość sterowanych za ich pomocą urządzeń. Podstawowymi cechami konstrukcyjnymi, uniemożliwiającymi uzyskanie dużej szybkości działania tych silników, jest duża masa wirnika, wykonanego z materiału ferromagnetycznego oraz indukcyjność cewek wzbudzających, uniemożliwiająca szybkie narastanie prądu. W superszybkich silnikach krokowych zmieniono konstrukcję silnika, wykonując stojan w postaci magnesu trwałego, zaś wirnik wykonany jest z laminatu epoksydowego, na którym znajdują się „drukowane" uzwojenia silnika. Aby zmniejszyć indukcyjność uzwojeń, stosuje się dodatkowe pokrycie nabiegunników magnesu trwałego stojana, warstwą dobrze przewodzącą z miedzi lub srebra. Rozwiązanie to pozwala na wykonanie pojedynczego kroku w czasie kilku ms, jest jednak związane z koniecznością stosowania specjalnych metod doprowadzenia prądu do uzwojeń wirnika. Stosuje się w tych wypadkach pierścienie ślizgowe lub, o ile silnik nie musi wykonywać pracy w zakresie zbyt dużego kąta, co ma miejsce np. w silnikach pozycjonujących głowice w jednostkach napędowych dysków, doprowadzenia elastyczne.

Oprócz typowych silników krokowych, które pozwalają na sterowanie urządzeń wymagających jako sygnału wyjściowego przesunięcia kątowego, istnieje możliwość budowy liniowych silników krokowych. Liniowy silnik krokowy przedstawiony jest na rysunku „c".

Wirnik zastąpiony jest tutaj magnesem trwałym, zaś stojan uległ rozwinięciu w liniowy system nabiegunników. Na stojanie znajduje się system dwóch uzwojeń, przy czym każde z nich obejmuje co drugi nabiegunnik stojana. Magnes ma możliwość przemieszczania się nad stojanem, równolegle do jego powierzchni. Zmiana kierunku prądu w poszczególnych uzwojeniach, identyczna jak w opisanym powyżej tradycyjnym silniku krokowym, powoduje ruch magnesu w jedną lub drugą stronę. W przypadku liniowych silników krokowych podstawową trudność sprawia ich łożyskowanie, znacznie bardziej skomplikowane niż w przypadku silników tradycyjnych, co znacznie podnosi koszty ich wytwarzania i sprawia, że nie znalazły one szerszego zastosowania.

Piotr Postawka

„Młody Technik — InforMik" wydaje Instytut Wydawniczy „Nasza Księgarnia"

Rada Redakcyjna: doc. dr Zygmunt Dąbrowski, inż. Jerzy Jasiuk, dr Zygmunt Kalisz, mgr Zbigniew Słowiński, mgr inż. Jerzy Siek, dr Zbigniew Płochocki, Piotr Postawka, mgr inż. Roland Wacławek, prof. dr hab. Andrzej K. Wróblewski (przewodniczący), mgr inż. Grzegorz Zalot.

Zespół redakcyjny: *„InforMik"* redaguje zespół *„Młodego Technika"* — Jerzy Klawiński (sekretarz red.), Jacek Nowicki (red.), Dariusz A. Przygoda (red.), Lidia Sadowska-Szlaga (korekta), Józef Trzionka (redaktor naczelny), Roland Wacławek (software), Grzegorz Zalot (hardware), Izabella Żur (red. tech.).

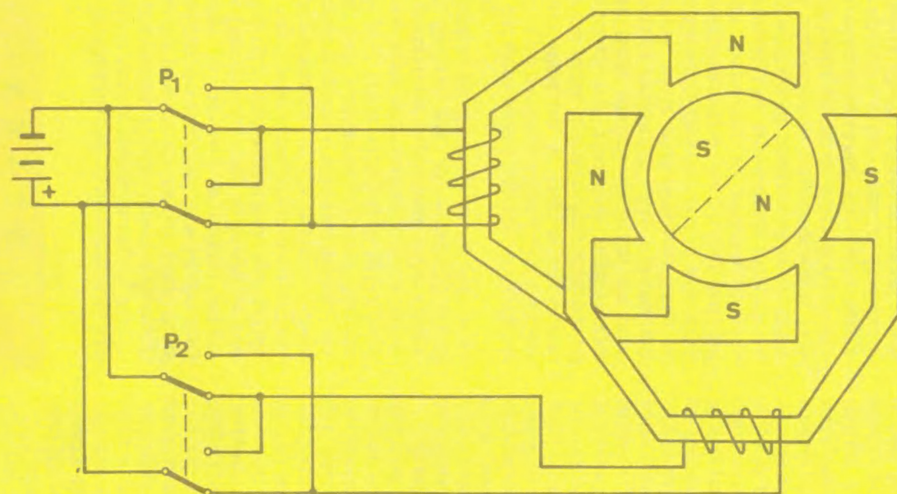
Stali współpracownicy: Wojciech Apel, Tadeusz Basista, Jacek Jędrzejowski, Piotr Postawka, Marek Szczepański, Krzysztof Wiśniewski.

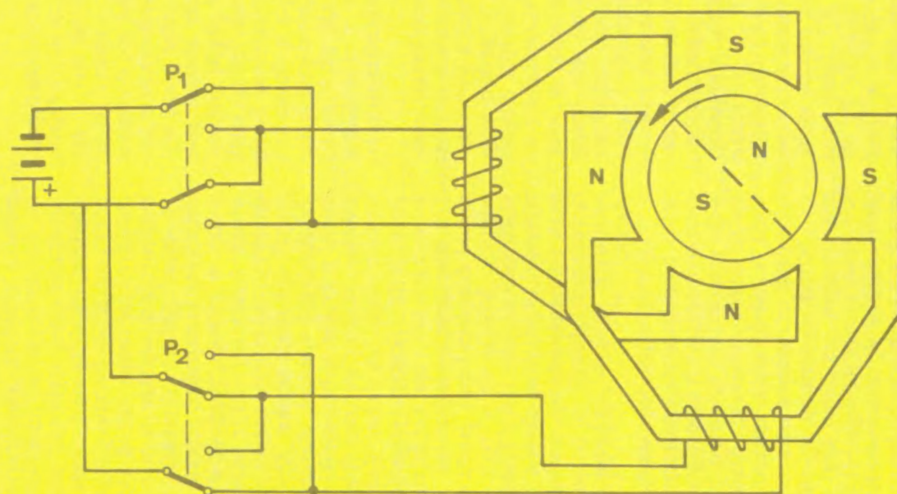
Adres redakcji: ul. Spasowskiego 4, 00-389 Warszawa, lub skr. poczt. 380, 00-950 Warszawa. **Telefony:** centrala: 26-24-31 do 36. Dział Łączności z Czytelnikami — wewn. 60, pozostałe działy: wewn. 42 i 47. Redaktor naczelny: 26-26-27 lub wewn. 87.

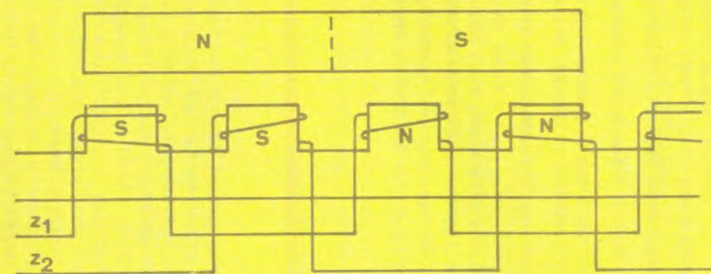
Warunki prenumeraty: ogólnie obowiązujące w kraju.

Redakcja zastrzega sobie prawo adiustacji i skracania nadesłanych materiałów. Artykułów nie zamówionych redakcja nie zwraca.

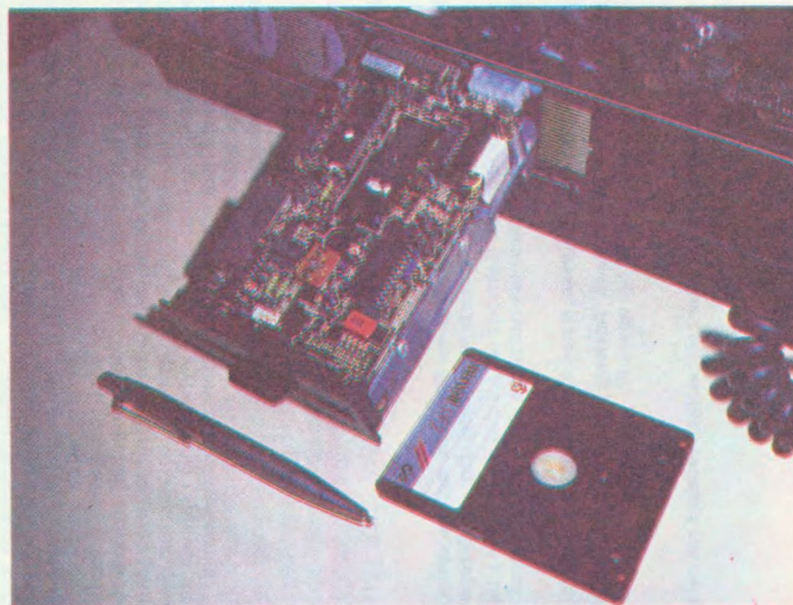
Druk: Zakłady Graficzne w Katowicach. Zam. 1516/4333/8 U-5
Nakład 60 000 + 315 egz.







Cena zł 120,—



Indeks nr 366013 PL ISSN-0860-5696